1984-03

Mean flow and turbulence in complex terrain NPS/Vandenberg measurement system

Schacher, G. E. (Gordon Everett)

Monterey, California. Naval Postgraduate School

NPS-61-84-005

NAVAL POSTGRADUATE SCHOOL

Monterey, California

MEAN FLOW AND TURBULENCE IN COMPLEX TERRAIN
NPS/VANDENBERG MEASUREMENT SYSTEM

by

G. E. Schacher and T. P. Stanton

March 1984

Approved for public release; distribution unlimited

Prepared for: U.S. Air Force, Space Division
Los Angeles, California 90009

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Commodore R. H. Shumaker
Superintendent

D. A. Schrady

The work reported herein was supported in part by U.S. Air Force Space Division, Los Angeles, California.

This report was prepared by

G. E. Schacher
Professor of Physics

T. P. Stanton
Professor of Oceanography

Approved by:

G. E. Schacher, Chairman
Department of Physics

C. N. K. Mooers, Chairman,
Department of Oceanography

Released by:

J. N. Dyer, Dean of
Science and Engineering

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS-61-84-005	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Mean Flow and Turbulence in Complex Terrain NPS/Vandenberg Measurement System		5. TYPE OF REPORT & PERIOD COVERED August 1983 - March 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Gordon E. Schacher and Timothy P. Stanton		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Code 61Sq Monterey, California 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS FY76168800471
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Air Force Space Division Code SD/CFA Los Angeles, California 90009		12. REPORT DATE March 1984
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

complex terrain
diffusion
turbulence

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A data acquisition system has been installed at Vandenberg Air Force Base to obtain meteorological data from 12 of the existing 23 towers. Data processing and storage includes means and standard deviations for time periods of 15 sec, 5 min, 15 min, and 1 hr. The 15 sec data contains those quantities needed to form averages, with standard deviations,
(continued)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Item 20. (continued)

over any longer time period. All data contains turbulence frequency components to 1 Hz. The purpose of the project is to characterize turbulence in complex terrain for use in diffusion modeling.

MEAN FLOW AND TURBULENCE IN COMPLEX TERRAIN VANDENBERG/NPS MEASURING SYSTEM

by

G. E. Schacher and Timothy Stanton

ABSTRACT

A data acquisition system has been installed at Vandenberg Air Force Base to obtain meteorological data from 12 of the existing 23 towers. Data processing and storage includes means and standard deviations for time periods of 15 sec, 5 min, 15 min, and 1 hr. The 15 sec data contains those quantities needed to form averages, with standard deviations, over any longer time period. All data contains turbulence frequency components to 1 Hz. The purpose of the project is to characterize turbulence in complex terrain for use in diffusion modeling.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

Major funding for this project was provided by the U.S. Air Force, Space Division; additional funding was provided by the U.S. Army Atmospheric Science Lab and the Naval Postgraduate School. The enthusiasm of Dr. Raymond Bernberg of Aerospace Corporation contributed substantially to making this project possible. The cooperation of Vandenberg AFB personnel was essential in getting the project underway. Special thanks are due Mr. Jerry Farley and Mr. Robert Harris of Federal Electric Company and Lt. Col. William Bihner and his staff meteorologists. Dr. William Ohmstede of the Atmospheric Science Lab gave valuable advice on the design of the project.

I. INTRODUCTION

Complex terrain meteorology is one of the more important and least understood issues in air pollution research. For example, a plume intersecting a hillside could produce very high local concentrations at the impacted face or could be channeled around the hill by the local airflow with resulting low concentrations. Another critical question is the manner in which complex terrain affects turbulence and hence, dispersion. It is reasonable to expect that airflow over irregularities in terrain will experience enhanced turbulence, but this is not necessarily so when compressed flow and channeling are involved. Which of the many effects, channeling, turbulence enhancement, etc., are most important for given conditions and the magnitudes of the effects can only be answered by a concerted program that includes both modeling and measurement.

Field data of the type that can be used to answer these important questions are largely inadequate due to the locations in which the data were obtained, the types of data acquired, and the data collection methods. The project described in this report should supply a great deal of the data needed to attack this important problem.

There have been several good measurement programs carried out in complex terrain, with perhaps the best being the Atmospheric Studies in Complex Terrain, ASCOT, project, coordinated by the Lawrence Livermore National Laboratory. The purpose of ASCOT is to study nocturnal drainage flows. The ASCOT program concentrates on intensive measurement periods of fairly short duration. Such a program, while important, does not accomplish the goals addressed here.

The purpose of the program described here is to develop a statistically valid data base to characterize the mean flow and turbulence in complex terrain for a wide range of conditions and terrain types. In order to do so, data will be collected for a full year, utilizing sensors distributed over a wide area at Vandenberg Air Force Base.

The Army, Air Force, Navy, and civilian sectors all have programs to model diffusion in complex terrain. Data obtained in this project will be incorporated into the DoD programs. Current users of the data which have already been delivered are the Air Force Space Division for Space Shuttle impact modeling, the Army Atmospheric Science Lab for validation of 3-D flow models, and the Naval Postgraduate School, NPS, for parameterizing coastal diffusion models and a complex terrain puff model.

Vandenberg Air Force Base is one of the few areas in the world where a project of this type could be carried out. It has 23 meteorological towers permanently installed in various types of terrain. The base is on the Pacific Coast and includes both high coastal hills and flat areas, making it ideal for both coastal and complex terrain studies. All data is available at a central location so that installing the needed data acquisition system can be accomplished. Figure 1 shows the location and the general features of the base. A more complete description of the location, climatology, and permanently installed equipment is contained in the following sections.

Since the data is being collected at Vandenberg, it will be most useful for developing site-specific models. However, the terrain is varied enough and enough conditions will be encountered over a one year period that the data will be applicable for general model development. Of course, one of the major uses of the data will be to develop a model to assess the impact of the Space Shuttle exhaust.

One of the more important aspects of this program is to be able to determine turbulence intensities over the full range of scales that contribute to dispersion of a plume or puff of material. As will be seen in what follows, the data contains the information needed to assess turbulence scales from a few meters to mesoscale size eddies (frequencies from 1 Hz to many hours).

The following sections outline, in more detail, the project strategy. General descriptions of the Vandenberg meteorological system are presented in Section II. Section III details specific meteorological site characteristics, and Section IV contains the area climatology. The Vandenberg data acquisition system is described in Section V and the NPS system in Section VI. The NPS methodologies for data acquisition, processing, and analysis are contained in the remainder of the report.

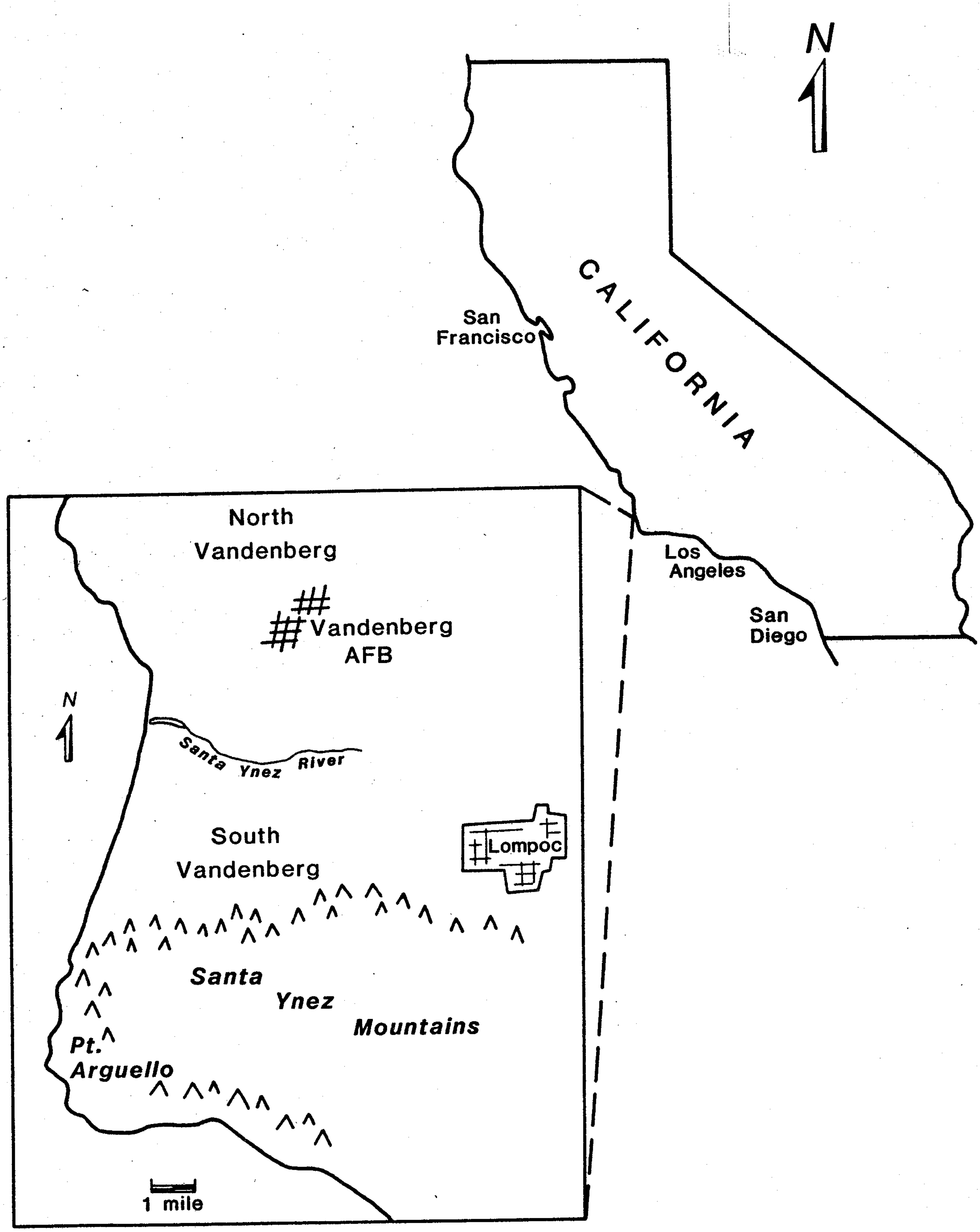


Figure 1. Geographical Location of Vandenberg Air Force Base and General Features

II. GENERAL DESCRIPTION OF THE VANDENBERG METEOROLOGICAL SYSTEM

The Vandenberg meteorological system consists of 23 towers of varying height distributed over an area approximately 10 by 20 miles on a side. The shortest towers are 12 feet and the tallest 300 feet. All of the towers that are more than 12 feet high have instrumentation at more than one elevation. The available instruments are:

1. wind speed,
2. wind direction (horizontal component),
3. air temperature,
4. dew point temperature,
5. barometric pressure,
6. visibility,
7. long and short-wave radiation,
8. rain rate.

Items 5 through 8 are included at only one site on South Vandenberg and item 4 at only two sites, one on the north section, and one on the south. Details of the types and the characteristics of the sensors are shown in Table 1.

Wind sensors are installed at heights of 12, 54, 102, 204, and 300 feet (the few exceptions are noted in the next section). Temperature sensors are at the same heights, with the exception of the lowest height which is 6 rather than 12 feet. Pressure, visibility, radiation, and rain rate are all measured at 6 feet.

SENSOR	TYPE	CHARACTERISTICS
Temperature	Rosemount Series 78 platinum resistance in Geotech Model 327 aspirated radiation shield	accuracy ± 0.3 deg C
Dew Point	EG&G Model 110S-M cooled mirror system	Range -80 deg F to 120 deg F accuracy ± 0.5 deg F
Wind Speed	Geotech 1564B with 170-41 plastic cups	threshold 0.63 mph distance constant 5 ft accuracy ± 1%
Wind Direction	Geotech 50.2C with 53.1	threshold 0.7 mph at 10 deg distance constant 5.7 ft turning radius 29 in damping ratio 0.4 at 10 deg
Visibility	MRI 1580 A visiometer	range 0-50 mi
Pressure	Yellow Springs, Inst. Series 2014, bellows type	range 909-1073 mbar accuracy ± 0.5 mbar
Short wave	Epley PSP precision pyranometer	range 0-2 Langley accuracy ± 0.1 Langley

Table 1. Vandenberg Meteorological Sensor Characteristics

The data obtained and recorded are the measured values from the sensors, with the exception of the elevated temperatures. The differences in the temperatures at the upper levels and those at the lowest levels are determined and recorded. This feature is desirable because it removes large errors that could occur in determining temperature gradients if the differences were calculated from the noisy data after transmission.

The system samples data every second and averages that data for 15 minute periods. Means for all data are calculated and the wind direction standard deviations are also computed. Other quantities that are not of interest here, such as gust statistics, are also determined. Data are output to 9-track tape and teletype. The tapes are archived while the teletype print-outs are saved for one month, then discarded. Details about the archived data and its availability can be obtained from Vandenberg Personnel. A reproduction of the 15 minute print-out is shown in Figure 2.

2059	PST	8	4	84				
0459	ZULU	9	4	84	15min ave	DISC-0	FILE- 25	

NORTH VANDENBERG

SITE	LVL	WS	WD	SIGMA	T-6´T
052	12	11	348	12.1	45.7
	54	15	343	11.2	2.0
102	12	8	330	8.1	57.2
	54	12	317	39.5	3.3
	102	21	330	4.7	

SOUTH VANDENBERG

SITE	LVL	WS	WD	SIGMA	T-6T	DPT	VIS06	BAR06
009	12	0	000	0.0				
014	12	15	037	25.9				
054	12	14	341	7.4	50.9			
	54	13	335	14.2	-1.6			
055	12				45.4			
	40	32	346	9.4	9.9			
056	40	16	029	25.1	42.8			
101	12	15	355	7.8	45.7			
	54	16	344	6.9	1.4			
103	12	0	000	0.0	120.0			
	54	0	000	0.0	-10.0			
200	12	11	343	13.5	48.1			
	54	18	338	8.0	2.5			
	102	20	344	5.9				
	204	26	337	4.9	-.3			
299	108	33	339	5.3				
300	12	14	344	8.8	49.3			
	54	18	342	7.2	.2			
	102	21	335	5.9	-.6			
	204	21	327	4.6	-1.0			
	300	18	341	3.4	-2.1			
301	12	15	350	11.4	53.0	45.9	.8	1011.2
	54	21	004	8.2	0.0	46.2		
	102	23	356	5.8	.6	45.1		
	204	25	352	6.8	-.6	38.8		
	300	27	359	0.0	-1.6	-90.0		

SW = .06

Figure 2. Sample of the 15 Min. Print-Out

III. METEOROLOGICAL SITES AND SENSORS

III-1 GENERAL DESCRIPTION

The sites for data collection were chosen so as to include as wide a range of terrain types as possible, while concentrating the locations on South Vandenberg. Two of the twelve sites are on North Vandenberg and the remaining nine are on the South section. There is a line of sites within one mile of the shoreline, a site in a broad river valley, and sites that are in the hills well behind the shoreline. Figures 3 and 4 are contour maps of Vandenberg showing the locations of the sites.

Four of the sites are within 1/2 to 1 mile from shore, 102, 200, 300, and 301. The northernmost of the four is 102 which is in a relatively flat area composed mainly of vegetated sand dunes. The other three are located inland of a fairly high shoreline bluff.

Sites 009, 052, and 102 are on the flattest terrain, 009 being the one on the river bed. 052 is next to the air field (at North Vandenberg) but far enough from it so that the terrain is slightly rolling.

The terrains behind the bluff at the three sites closest to the beach on South Vandenberg are somewhat rolling but fairly flat. The inland hills rise fairly sharply behind these sites. Site 200 is on the edge of a major canyon that extends 4 miles into the hills.

Site 103 is a special case. It is next to Santa Barbara Channel, shielded by Point Arguello. The immediate locality is

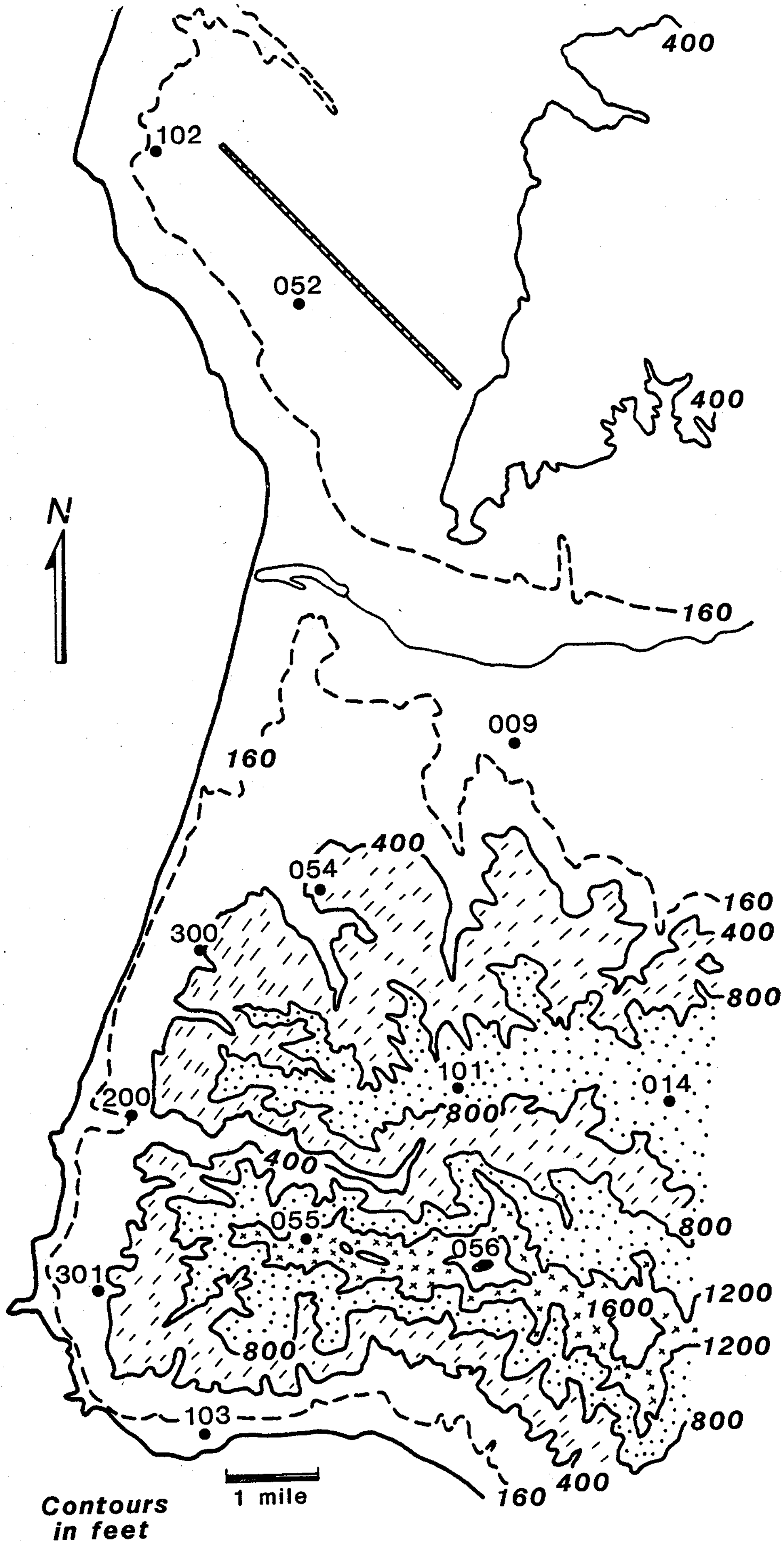


Figure 3. Vandenberg Topography and Meteorological Tower Sites

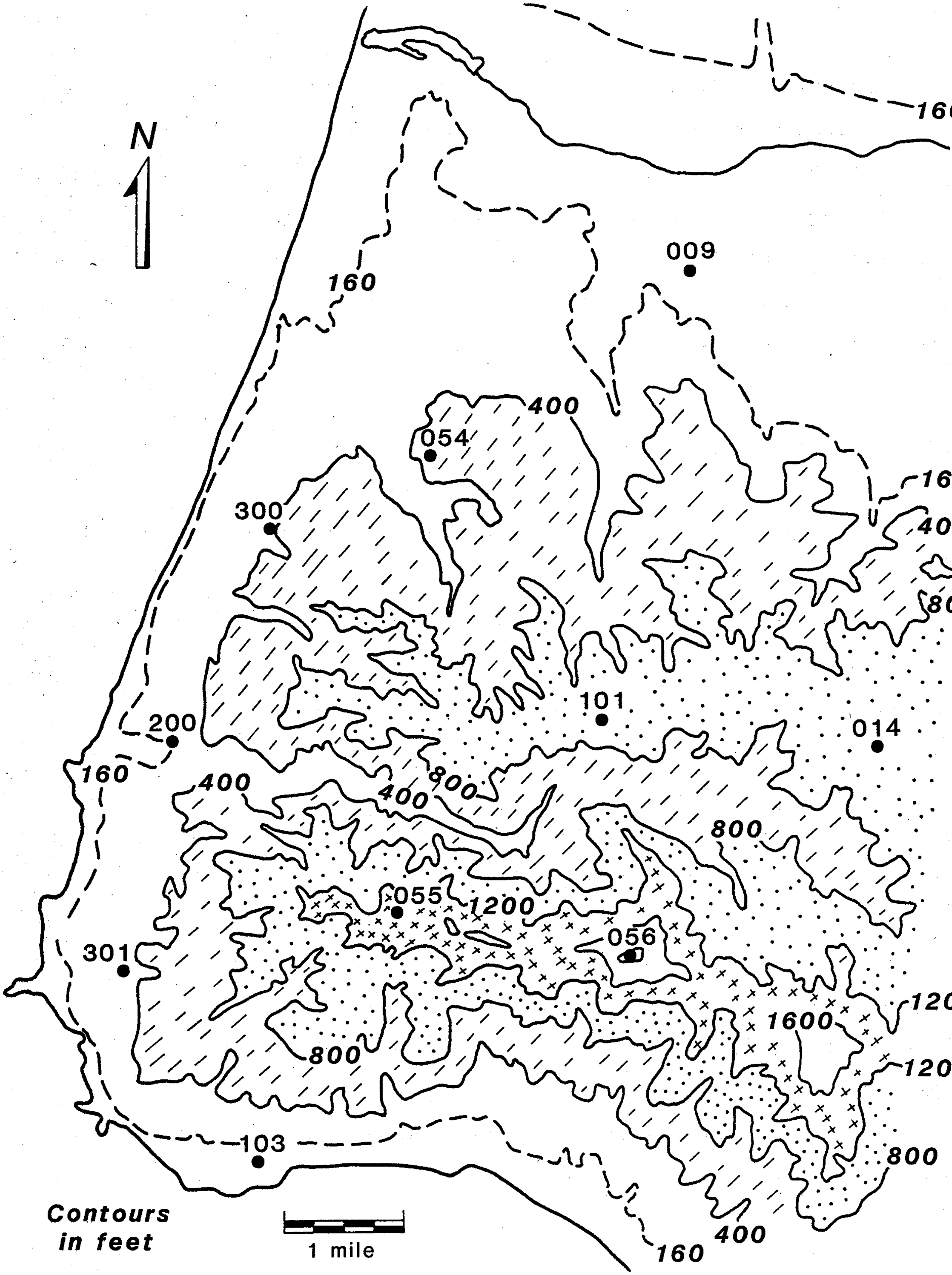


Figure 4. South Vandenberg Topography and Meteorological Tower Sites

much the same as the other near shore sites but the influence of Point Arguello is so great that the characteristics of the wind should be completely different.

The remaining sites are in the coastal hills. Some of these are in locations that are disrupted by structures which are located nearby. Complete descriptions of the terrain around the individual sites and the characteristics of the meteorological equipment follow in this section.

III-2 TERRAIN AT THE INDIVIDUAL SITES

This section describes both the types of terrain at each of the sites and any man made structures that can affect the local airflow. The descriptions are fairly condensed since it would be nearly impossible to describe all features which could influence the wind. The purpose here is to give a generic description of the locality that can be used to classify it and also to detail those features that could produce flow disruptions.

There are two types of vegetation which dominate the project area: coastal sage scrub and grassland. Trees in the area are not located close enough to any of the sites to be a significant factor.

Each site includes a tower 6 feet by 4 feet, which is constructed of 2 inch pipes, except for those which are less than 50 feet high which are small triangular meteorological towers. The height of a tower can be determined by the number of the tower. If the number is less than 050, the height is less than 50 feet. All towers are accompanied by a small electronics shelter which is about 50 feet away from the base of the tower and is about 5 feet square and 8 feet high. The shelters are located so that they do not influence the flow to the towers when the wind is from the predominant direction. Also, the sensors on the towers are placed so that they are not shadowed by the tower when the wind is from the predominant direction.

009	Elevation:	27 ft
	Terrain:	flat river flood plane
	Vegetation:	plowed field with grass at tower base, treeline 100 yds to N
	Structures:	electronic shelter to SW, 30 ft high building 100 yds SE
014	Elevation:	1446 feet
	Terrain:	gentle slopes to all sides, 20 ft to canyon S of tower, ridge 1-2 mi W
	Vegetation:	low grass and rock
	Structures:	10 foot shed 30 ft NE
052	Elevation:	290 ft
	Terrain:	flat
	Vegetation:	grass 1-2 ft
	Structures:	2 low huts 300 yds SW, shelter E of tower
054	Elevation:	450 ft
	Terrain:	well-exposed with a gradual slope toward the W, canyon to immediate S and W W dropping off quickly, slight slope to N
	Vegetation:	grass and shrub
	Structures:	launch pads NW and SE of tower 150 ft apart, shelter located ESE of tower
055	Elevation:	1530 ft
	Terrain:	located on top of mountain peak, sharp drop-off 20 ft W, otherwise flat on top
	Vegetation:	concrete and dirt at site itself, otherwise surrounded by chapparal 2-5 ft in height
	Structures:	40 ft telephone pole immediately S, 60 ft radome 70 ft E of tower
	Comment:	Radome will cause severe flow distortion

056	Elevation:	2136 ft
	Terrain:	located on top of mountain peak with sharp drop-off 10 ft around permiter of site
	Vegetation:	concrete and asphalt on pad, surrounded by chapparal shrubbery on sloping hillsides
	Structures:	obstructive 25 feet high buildings 30 ft to SW and NNW of tower, radar dish above SW building
	Comment:	high winds indicative of this site, temperature sensor at 6-foot level has possible heating influence from asphalt, some flow distortion from buildings
101	Elevation:	1077 ft
	Terrain:	tower on top of small knoll with gradual slope to W and N, 12 ft hill 50 ft SE, deep canyon 300 ft S, small canyon to immediate N
	Structures:	buildings on adjacent hill 1/4 mi NNW at same height as tower base
	Comment:	lower wind vane doesn't appear to be influenced by hill to SE
102	Elevation:	215 ft
	Terrain:	flat with gradual slope to beach front
	Vegetation:	grass and sand, 10-foot hill 60 ft E
	Structures:	launch pad 300 yds S, shelter 30 ft SE, two low buildings 400 yds to S
103	Elevation	56 ft
	Terrain:	in center of gentle NS slope, hills 1/4 mi N, coastline 1/4 mi S with 100-ft drop-off
	Vegetation:	grassland 2 ft high
	Structures:	shelter to S, boat house approx 20 ft high located 200 ft ESE
	Comment:	not in operation at start of project
200	Elevation:	310 ft
	Terrain:	immediate location flat, short period rolling terrain to W, deep canyon to immediate S banked by high hills
	Vegetation:	3-5 ft chaparral shrubs
	Structures:	20 foot building 200 ft NW
300	Elevation:	385 ft
	Terrain:	relatively flat, slight grade to N and E of tower, ending in gentle hills, slightly rolling terrain toward the W
	Vegetation:	3 ft chaparral shrubs
	Structures:	tower located between two 300 ft gantries 150 ft apart WNW and ESE of the tower

301	Elevation:	381 ft
	Terrain:	fairly flat, sharply rising hills one-half mi E
	Vegetation:	2-5 foot chaparral shrubs
	Structures:	four-story building 1/4 mi NW, major structures approx. 600 ft E

III-3 SENSORS INCLUDED AT EACH SITE

Data obtained include wind speed, wind direction, temperature at 6 ft, the temperature difference between the upper levels and the 6 ft level, and on tower 301 dew point, visibility, barametric pressure, and short-wave radiation. The data obtained at each elevation at each site are shown in Table 2.

With the exception of site 301, all of the data available from each site was collected. Rain gauge and long wave radiation information from 301 was not obtained because it is not of interest. Note that 299 is not an independent site but an extra level at 108 ft at site 300. It is included as a battery driven backup in case power is lost at the site.

Site Number	Sensor Height (feet) 6	12	40	54	102	204	300
009		WS WD					
014		WS WD					
052	TEM	WS WD		WS WD dT			
103	TEM	WS WD		WS WD dT			
054	TEM	WS WD		WS WD dT			
055	TEM		WS WD	dT			
056	TEM		WS WD				
101	TEM	WS WD		WS WD dT			
102	TEM	WS WD		WS WD dT	WS WD		
200	TEM	WS WD		WS WD dT	ws wD	ws wD dT	
299*					WS WD (108 ft)		
300	TEM	WS WD		WS WD dT	WS WD dT	WS WD dT	WS WD dT
301+	TEM DP	WS WD	~~WS WD~~ ~~dT DP~~	WS WD dT DP	WS WD dT DP	WS WD dT DP	WS WD dT DP

*Piggyback to site 300
+Also VI, BP, SW

TEM-Temperature
WS- Wind Speed
WD-Wind Direction
VI-Visibility
BP-Barometric Pressure
SW-Shortwave Radiometer
dT-Temperature Difference
DP-Dew Point

Table 2. Sensors and Their Elevations at Each Site

IV. VANDENBERG CLIMATOLOGY

IV-I GENERAL DESCRIPTION OF THE AREA

Vandenberg is located on the south-central California coast just north of the Santa Barbara Channel. The local climatology is largely controlled by two factors: the semi-permanent Pacific high and the moderating influence of the ocean. Because of the Pacific high, low temperature inversions and accompanying stratus and/or fog are a common occurrence, particularly from spring to fall. The ocean maintains a predominantly marine atmospheric boundary layer in the area. However, due to the high coastal hills, for some circumstances, there can be rapid changes in conditions with distance from shore.

The area is also subject to the "Santa Ana" conditions which frequently affect much of southern California. Under these conditions, dry, hot winds sweep out of the inland desert causing high temperatures, low humidity, and high winds all the way to and even beyond the coast. When a Santa Ana occurs, it is possible to have hurricane force winds in valleys and canyons.

For the vast majority of the time the climate at Vandenberg is temperate. Summers are cool due to the presence of the stratus and fog. Occassional temperatures of 90 deg F occur from August to October. Minimum temperatures are seldom below 40 deg F in the winter and 50 deg F in the summer.

The winter season rainfall is modest, averaging about 12 in from November to April. The summers are dry, May to October rainfall averaging 1 in.

Being a coastal site, the area experiences the diurnal land-sea-breeze cycle due to the differential in the heating between the land and the sea. The sea breeze frequently reaches speeds of 15-20 kts. The breeze dies off at dusk and flow reversal can occur in the middle of the night. Night time downslope drainage from the surrounding hills tends to strengthen the land-sea-breeze cycle.

The sea-breeze cycle is superimposed on the gradient wind so that it may be masked when the pressure gradient is large, being evidenced only by a variation in the speed of the onshore breeze. When stratus is present, the differential heating is small due to the reduction in the solar radiation reaching the surface. Under these conditions a steady onshore flow can continue for days with little change.

A diurnal variation in the fog and stratus occurs when there is a near balance in the differential heating and the maintenance of the marine boundary layer and stratus by the inversion. Fog will "roll in" during the late afternoon and remain until the following mid-morning when it will be burned off to as far as several miles off shore. This cycle is quite common during the summer.

During the winter season, conditions are much more variable with occasional storms sweeping through the area, with frontal activity and strong winds following frontal passage.

The above highlights indicate that the Vandenberg location is subject to a wide range of influence, which are often in competition. The resulting meteorological conditions vary over a wide range, especially when geographical differences in the data collection sites are also considered. Seasonal descriptions of coastal weather conditions and specific Vandenberg meteorological conditions are given in the remainder of this section.

IV-2 CALIFORNIA COAST SEASONAL CLIMATOLOGY

SUMMER

The North Pacific semipermanent subtropical high lies to the west of the area and controls the synoptic scale flow. Clockwise flow around the high produces northwesterlies along much of the coast, with the local sea-breeze turning the wind more westerly. The general onshore flow is aided by the inland thermal trough which is created by overland heating. Strong subsidence creates the prevalent capping inversion and the occasional passage of weak upper level troughs will dissipate or lift the inversion for periods of 12-24 hours.

FALL

The building of high pressure in the Great Basin causes frequent Santa Ana conditions. The pattern of storms and upper level westerlies moves further south breaking up the summer pattern. Frontal passage becomes more frequent and the subtropical high becomes displaced or shrinks, resulting in a break up of the marine inversion.

WINTER

Frontal passage becomes much more frequent and strong surface westerlies often follow the passage. Santa Ana winds can still occur when the surface pressure in the great basin becomes sufficiently high. Also, the Pacific High and capping inversion can reform between frontal passage occurrences.

SPRING

As the storm pattern moves north, the Pacific High again becomes the dominant feature. Cold lows pass frequently, followed by strong westerlies.

IV-3 VANDENBERG WIND CLIMATOLOGY

Wind climatologies are useful in determining expected conditions and for assessing whether observed conditions are typical. It is not possible to use the climatology to accurately predict local conditions on a day by day basis but seasonal patterns are quite reproducable. Surface wind roses for the area, taken from a site near the air field are shown in Figure 5.

Since Santa Ana conditions are typically of short duration; the wind roses do not sufficiently account for them. These situations are so atypical that it is useful to know how frequently they occur. The average number of days of Santa Ana occurrences for the 12 months from January through December are:

Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
9.3	5.2	2.8	0.6	0.3	0	0	0	0.4	2.7	7.0	9.3

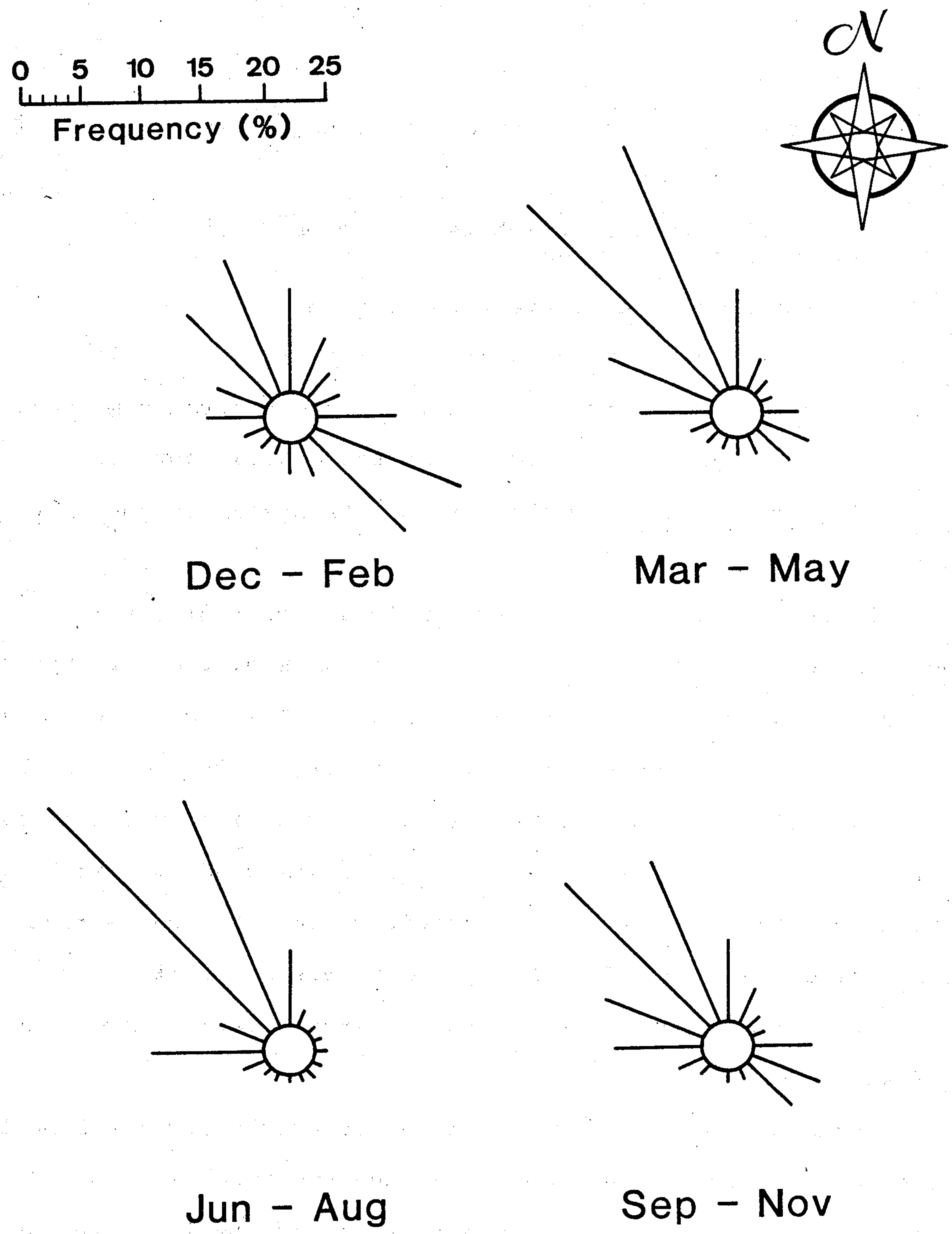


Figure 5. Vandenberg Monthly Wind Roses

V. VANDENBERG DATA ACQUISITION SYSTEM

V-1 GENERAL DESCRIPTION

The existing Vandenberg data acquisition system consists of the following major components:

1. a central computer which issues commands and interprets data words,
2. analog to digital converters at each site,
3. transmit and receive modems at each site,
4. hard-wire lines from each site to the central computer,
5. transmit and receive modems at the central computer.

A block diagram of the basic elements of the system is shown in Figure 6.

The basic operation of this system is as follows: the computer issues a command word that calls for a particular type of data, such as wind speed at the 54 ft level. The command word is converted to a frequency-shift-keyed (FSK) signal by the transmit modem. This signal is sent to all sites on the line, where it is translated by the receive modems. All sites respond by switching the requested data to the sites transmit modem. At the central station the computer decides which site it wishes to listen to and switches the line from that site to the central receive modem by use of selection relays.

The systems at each end of the line are coupled to the line by hybrids which separate the transmitted and received signals. A line equalizer is used to match the receiver to the line. Because

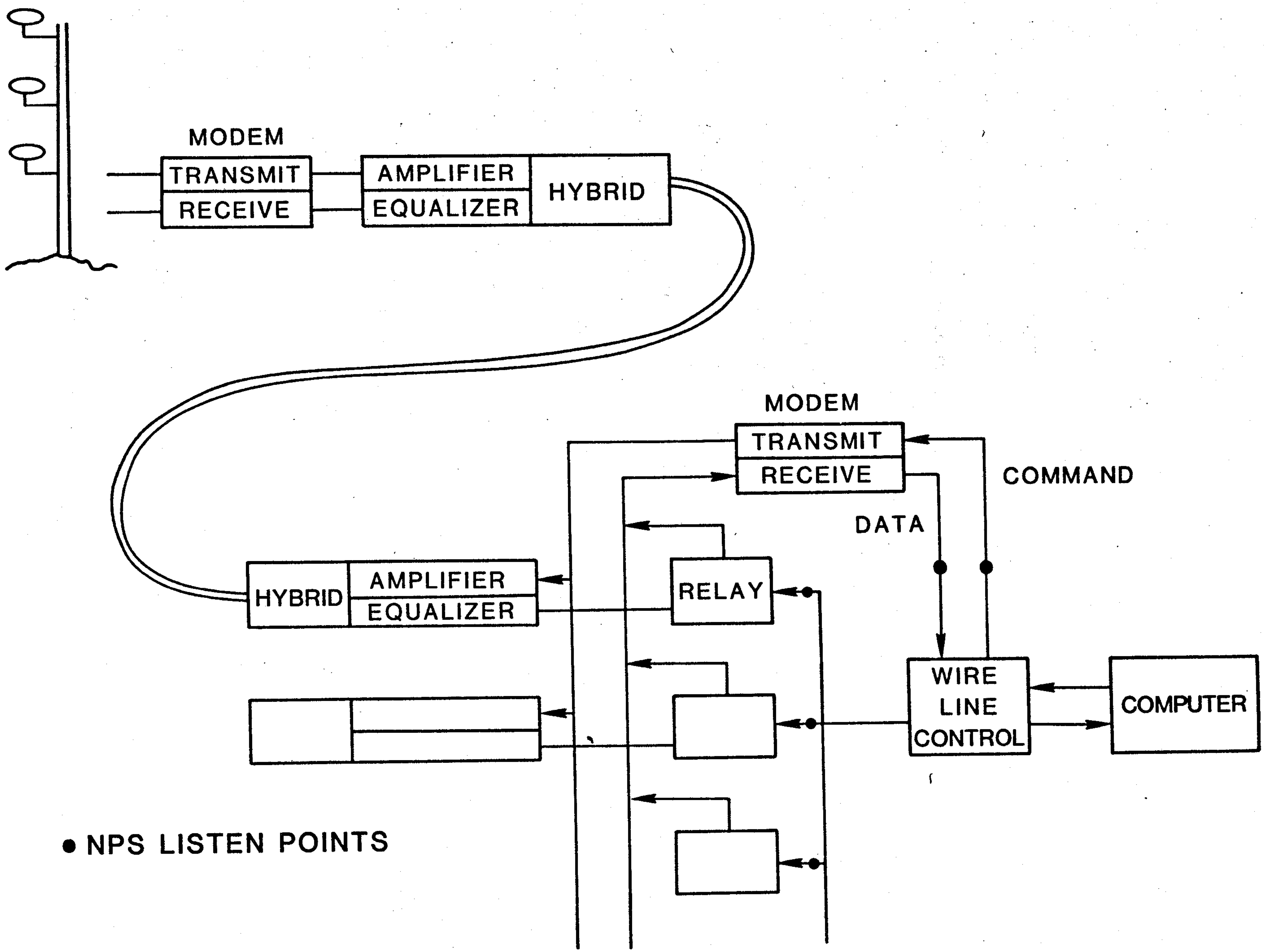


Figure 6. Block Diagram of the Vandenberg Data Acquisition System

the cables are several miles long, line amplifiers are needed for all transmitted signals.

Using a predetermined sequence, all sensors on all towers are cycled through in approximately 0.8 sec. The time required to acquire data from any given sensor is approximately 20 msec. It takes the computer approximately 0.2 sec to process the data acquired in a cycle, giving a total cycle time of 1.0 sec, so that each sensor is read once each second.

A complete transceiver system, as described above, is needed at each site. Some equipment economy is realized at the central site by combining the data lines into four "wire-lines", each of which has its complete transceiver system. Due to the evolution of this system over many years, the wire lines are not identical, as will be seen below. The computer operates the four wire-lines almost simultaneously, with a 0.08 msec delay between command words to successive lines. Wire-lines 1 and 2 have the same characteristics, as do wire-lines 3 and 4. The basic characteristics are:

wire-lines	low freq	high freq	baud rate
1&2	90 Hz	8200 Hz	2400
3&4	1800 Hz	2400 Hz	2400

In the remainder of this section, only those details of the Vandenberg system that are necessary for understanding the NPS system will be described. This will include such factors as data coding, sequencing, etc.

V-2 COMMAND AND DATA WORDS

The binary command and data word structures are shown in Figure 7. The most significant aspect of these words is that they are not the same for all wire-lines. Wire-lines 1 and 2 have the same words, as do lines 3 and 4. Briefly, the differences are: the command word for wire-lines 3 and 4 has a calibration word included which is not present for 1 and 2. The least significant bit in the data word has special uses which are not the same for all lines.

The first bit of the command word is the sync bit and the second the parity bit. These are followed by 5 bits which contain the code for calling up the various sensors. The binary words and their decimal equivalents are shown in Table 3. Three additional bits are included in the command word for wire-lines 3 and 4 for the purpose of calling for a system calibration. Bit 8 is used to call for a high calibration (maximum sensor values), bit 9 for low calibration (intermediate sensor values), and bit 10 is the calibration word parity. The calibration word information is shown in Table 4.

		Bit	
Calibration	8	9	10
hi-cal	1	0	1
cal off	0	0	0
lo-cal	0	1	1

Table 4. Wire-lines 3 and 4 Calibration Command Words

Sensor	Decimal	Binary
WD-12	0	000000
WS-12	1	1
T-6	2	10
dT-54	3	11
WD-54	4	100
WS-54	5	101
WD-108*	6	110
WS-108*	7	111
WD-12+	8	1000
WS-12+	9	1001
WD-102	10	1010
WS-102	11	1011
WD-204	12	1100
WS-204	13	1101
DP-6	14	1110
dT-102	15	1111
Vis	16	10000
Bar	17	10001
WD-300	18	10010
WS-300	19	10011
dT-204	20	10100
dT-300	21	10101
DP-54	22	10110
DP-102	23	10111
DP-204	24	11000
DP-300	25	11001
	26	11010
SW	27	11011
LW	28	11100
Rain	29	11101

*Piggyback tower 299 only
+Piggyback tower 014 only

Table 3. Sensor Call-up Command Code Words

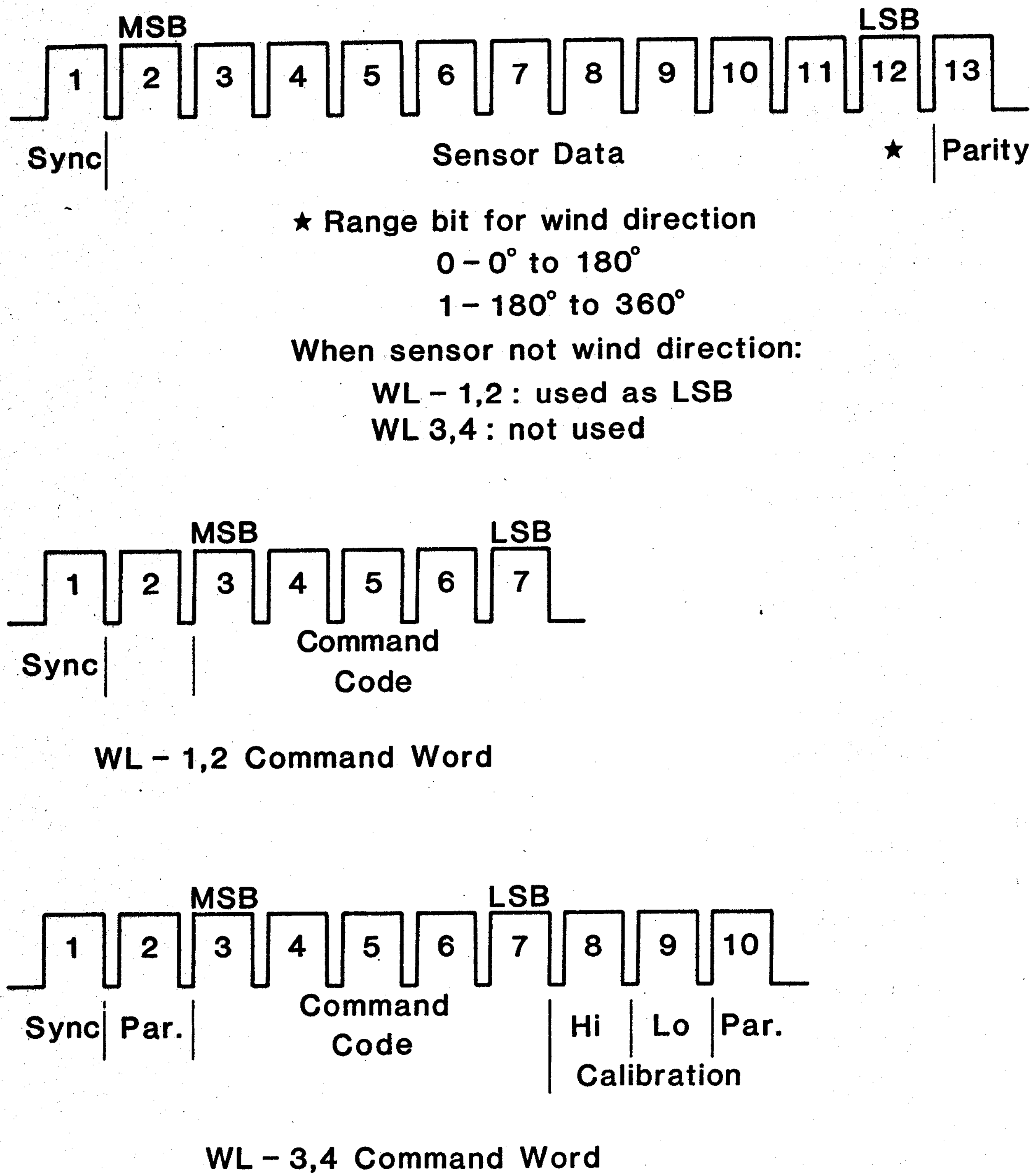


Figure 7. Command and Data Word Structures

Calibrations of the total system are performed periodically. This can be accomplished either by going to the site and manually setting a calibration value or by calling for the predetermined hi or lo-cal with the calibration command word. Of course, the only method available for those sites on wire-lines 1 and 2 is to go to the site. The standard calibration values that are used for the system are listed in Table 5. Note that the calibration values used depend on the site and also that it is possible to use any value desired when calibration is performed by going to the site. However, those values listed in the table are the ones that will most often be encountered.

	Sites 015, 052, 059 102, 300, 301	All Other Sites
Wind Speed (kts)	0, 44-45	60
Wind Direction (deg)	0, 180, 360	40, 160, 200, 320
Temperature (deg F)	15, 105	60, 100
Temp Diff (deg F)	-3.75, 8.75	-5, 0
Dew Point (deg F)	0, 110	0, 110
Visibility (mi)	0	0
Bar. Press. (in Hg)	26.85, 31.68	26.85, 31.68
Short Wave (Lang)	0, 2	

Table 5. Vandenberg Standard Calibration Values

The first bit of the data word is the sync bit and the last is the parity bit. The sensor data is contained in 11 bits so that the maximum value is 2047. Thus, the maximum precision of the data received is 1 part in 2047, or approximately .05% of the full scale value for any sensor.

The least significant bit in the data word is used both for data and to indicate the wind direction range when wind direction is the called data. The bit is 0 when the range is 0 to 180 deg and is 1 when the range is 180 to 360 deg. Note that the most significant bit is 2^{10} regardless of the use of the least significant bit. This means that the unit bit is missing for wind direction. The least significant bit is used only for wind direction range on wire-lines 3 and 4. However, it is used for data for all other sensors on wirelines 1 and 2.

The values of the returned sensor-words as a function of the value of the sensed meteorological quantity are shown in Figure 8. These values are also listed in Table 6.

V-3 CALL-UP SEQUENCE

The system acquires data from all of the sensors at all of the sites in a predetermined sequence. Data from the four wire-lines are called-up almost simultaneously (an 0.08 msec offset per line) and we ignore that offset in the following description. Thus, for our considerations, sequencing is made up of two separate sequences, that for the sites and that for the towers. There is a standard sensor call-up sequence that is used for all sites, however, the full sequence is used only for site 301 since it is the only site that has a full compliment of sensors. For time economy, only those sensors which are present at a given site are called when that site is polled.

The full sensor call-up sequence is shown in Table 7. The site, the sensor, and the command word are shown, for each sensor in sequence, for one full cycle of the system. As stated above, the time allotted for each command word to be sent and the return data word to be received is approximately 20 msec. Examination of the table shows that there are 40 time steps in the sequence, which accounts for 0.8 sec of the 1 sec cycle time. The other 0.2 sec are devoted to computer processing of the data. Unfortunately, there are spurious signals on the lines during this dead time which adversely affect the NPS system, as will be described in a subsequent section.

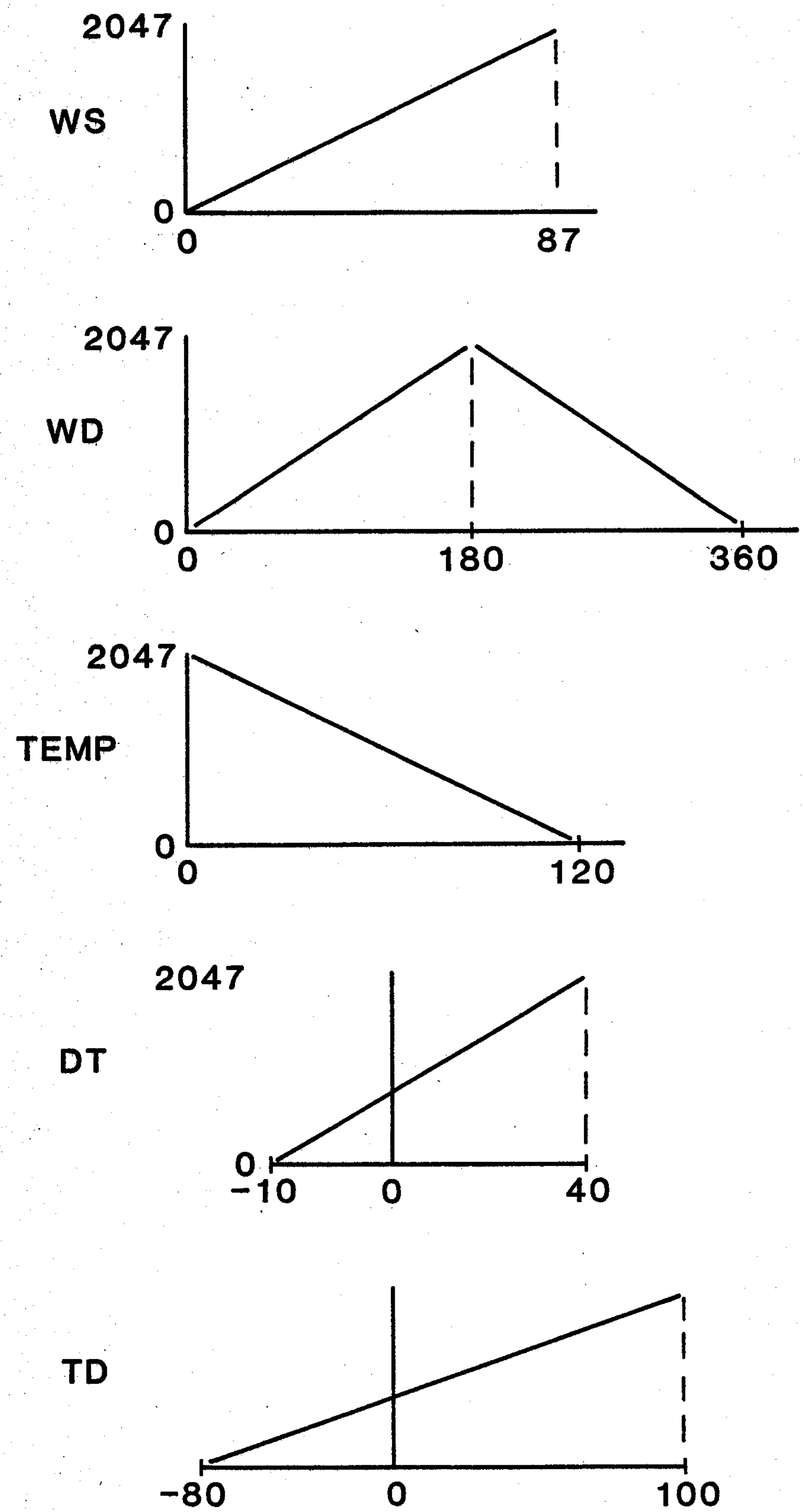


Figure 8. Sensor-word Value Versus Sensor Reading. WS = wind speed, WD = wind direction, Temp = air temperature, DT = temperature difference between elevated and 6 ft levels, TD = dew point.

Sensor	RANGE Min	RANGE Max	DECIMAL EQUIVALENT WL 1 & 2 Min	DECIMAL EQUIVALENT WL 1 & 2 Max	DECIMAL EQUIVALENT WS 3 & 4 Min	DECIMAL EQUIVALENT WS 3 & 4 Max	
Wind Speed U (Knts)	0	87	0	2047	0	2046	
Temperature T (°F)	0	120	0	2047	0	2046	
Delta Temp dT (°F)	-10	40	0	2047	0	2046	
Dew Point DP (°F)	-80	120			0	2046	
Wind Direction (deg)	0 180	180 360	0 2046	2046 0	0 2046	2046 0	(Bit 12 = 0) (Bit 12 = 1)
Visibility (mi)	0	50			0	2046	
Pressure (in Hg)	26.85	31.68			0	2046	
Short Wave SW(langley)	0	2			0	2046	

Table 6. Sensor Conversions

Wire Line - 1			Wire Line - 2			Wire Line - 3			Wire Line - 4		
Tower	Sensor	Code	Tower	Sensor	Code	Tower	Sensor	Code	Tower	Sensor	Code
050	WS12	1	058	WS12	1	015	WS12	1	052	WS12	1*
	WD12	0		WD12	0		WD12	0		WD12	0*
	T6	2		T6	2		T6	2		T6	2*
	WS54	5		WS54	5	102	WS12	1*		WS54	5*
	WD54	4		WD54	4		WD12	0*		WD54	4*
	dT54	3		dT54	3		T6	2*		dT54	3*
004	WS12	1	014	WS12	9*		WS54	5*	059	WS12	1
	WD12	0		WD12	8*		WD54	4*		WD12	0
005	WS12	1	054	WS12	1*		dT54	3*		T6	2
	WD12	0		WD12	0*		WS102	11*		BP	17
007	WS12	0		T6	2*		WD102	10*		DP 6	14
	WD12	0		WS54	5*	299	WS108	7*		WS54	5
017	WS12	1		WD54	4*		WD108	6*		WD54	4
	WD12	0		dT54	3*	300	WS12	1*		dT54	3
018	WS12	1	055	WS40	1*		WD12	0*		DP54	22
	WD12	0		WD40	0*		T6	2*	301	WS12	1*
009	WS12	1*		T6	2*		WS54	5*		WD12	0*
	WD12	0*		dT54	3*		WD54	4*		T6	2*
103	WS12	1*	056	WS40	1*		dt54	3*		DP6	14*
	T6	0*		WD40	0*		WS102	11*		Vis	16*
	WS54	2*		T6	2*		WD102	10*		BP	17*
	WD54	5*	101	WS12	1*		dT102	15*		WS54	5*
	dT54	4*		WD12	0*		WS204	13*		WD54	4*
				T6	2*		WD204	12*		dT54	3*
				WS54	5*		dt204	20*		DP54	22*
				WD54	4*		WS300	19*		WS102	11*
				dT54	3*		WD300	18*		WD102	10*
			200	WS12	1*		dT300	21*		dT102	15*
				WD12	0*					DP102	23*
				T6	2*					WS204	13*
				WS54	5*					WD204	12*
				WD54	4*					dT204	20*
				dT54	3*					DP204	24*
				WS102	11*					WS300	19*
				WD102	10*					WD300	18*
				WS204	13*					dT300	21*
				WD204	12*					DP300	25*
				dT204	20*					SW	27*
										LW	28*
										R	29*

*Sensors acquired by the NPS system

Table 7. Sensor Call-up Sequence

VI. NPS DATA ACQUISITION SYSTEM

VI-1 GENERAL DESCRIPTION

The operation of the Vandenberg data acquisition system was described in the former section. That system completely controls the data that appears on the wire-lines. The NPS system can not play an active roll in the data acquisition process without interfering with the Vandenberg system; it must be a passive listener. Figure 6 shows where the three needed signals are acquired, the command word, the data word, and the relay closure that tells which site is being polled. (Note that the clock signals for both command and data words are also acquired, but this is not shown in the figure.)

The command word is acquired directly at the computer output to the transmit modem. The data word is acquired after the FSK signal is translated into serial binary format by the receive modem. Site-relay activation is sensed by a separate connection to each of the relay drivers.

The components that make up the NPS system are:

1. Hewlett Packard 9826 computer,
2. Innovative Data Technology 9-track tape deck,
3. Hewlett Packard 82901 dual floppy disc drive,
4. Hewlett Packard 7281 thermal printer,
5. NPS constructed data acquisition unit,
6. 16-bit parallel interface.

All data processing, storage, and most of the acquisition control functions are performed by the mini-computer. The core of the computer has a capacity of 800 K, which means that it can perform only limited data storage functions. As much of the acquisition control logic as is possible is hard-wired into the acquisition unit to relieve the demands placed on the computer. As will be seen in the next section, many time saving "tricks" had to be employed in the programming of the system so that the single mini-computer could perform the needed tasks.

The data that are listened to are serial and the data from the four wire-lines arrive essentially simultaneously. All data are input to the computer through the single 16-bit interface, which requires multiplexing. The computer must read the command words, control words, and relay closures from 94 sensors in the 1 sec cycle time, and do this while it is also processing the data. This places a rather severe time and synchronization demand on the system. The means used to accomplish this are described in the next part of this section.

The output from the NPS system are means and standard deviations for averaging periods of 15 sec, 5 min, 15 min, and 1 hr. Not all averaging periods contain the same data, which will be explained in a later section.

Data are stored in two forms: all averages are stored on 9-track tape and the 15 min data are also stored on floppy discs. The purpose of the additional storage for the 15 min averages is to be able to easily make comparisons with the Vandenberg 15 min printed output. This is done for system performance checks.

Data output formatting and storage capacity are such that two full weeks of data can be stored on both media. Thus, the system runs unattended for that period. There are provisions for turning off the input from any site if there is a malfunction of the Vandenberg system. This can be done at any time by the Vandenberg personnel who maintain their system.

VI-2 NPS SYSTEM HARDWARE

The basic components of the NPS system are shown in the Figure 9 block diagram. The data from each wire-line is shifted into a tri-state latch using the Vandenberg system clock to strobe the data. The data are shifted out of the registers with the NPS computer clock, at which point they are multiplexed onto the data buss. There are many features which are not shown in this simplified diagram, such as the fact that the Vandenberg system has different clock signals for the command and data words and different clocks for the different wire-lines. The figure is not meant to be complete, only to show the basic configuration of the system.

In order for the NPS computer to have enough time to perform all of the required functions, data acquisition and processing must be performed almost simultaneously. Thus, the computer operates on an interrupt basis. When a site relay is closed and the shift registers are filled, an interrupt is generated and the computer shifts to the data loading mode. When loading of the data from all four wirelines is completed the computer immediately returns to the processing mode until the next interrupt. The interrupt priority is high, so that it will take place at the end of the line of code that is currently being processed.

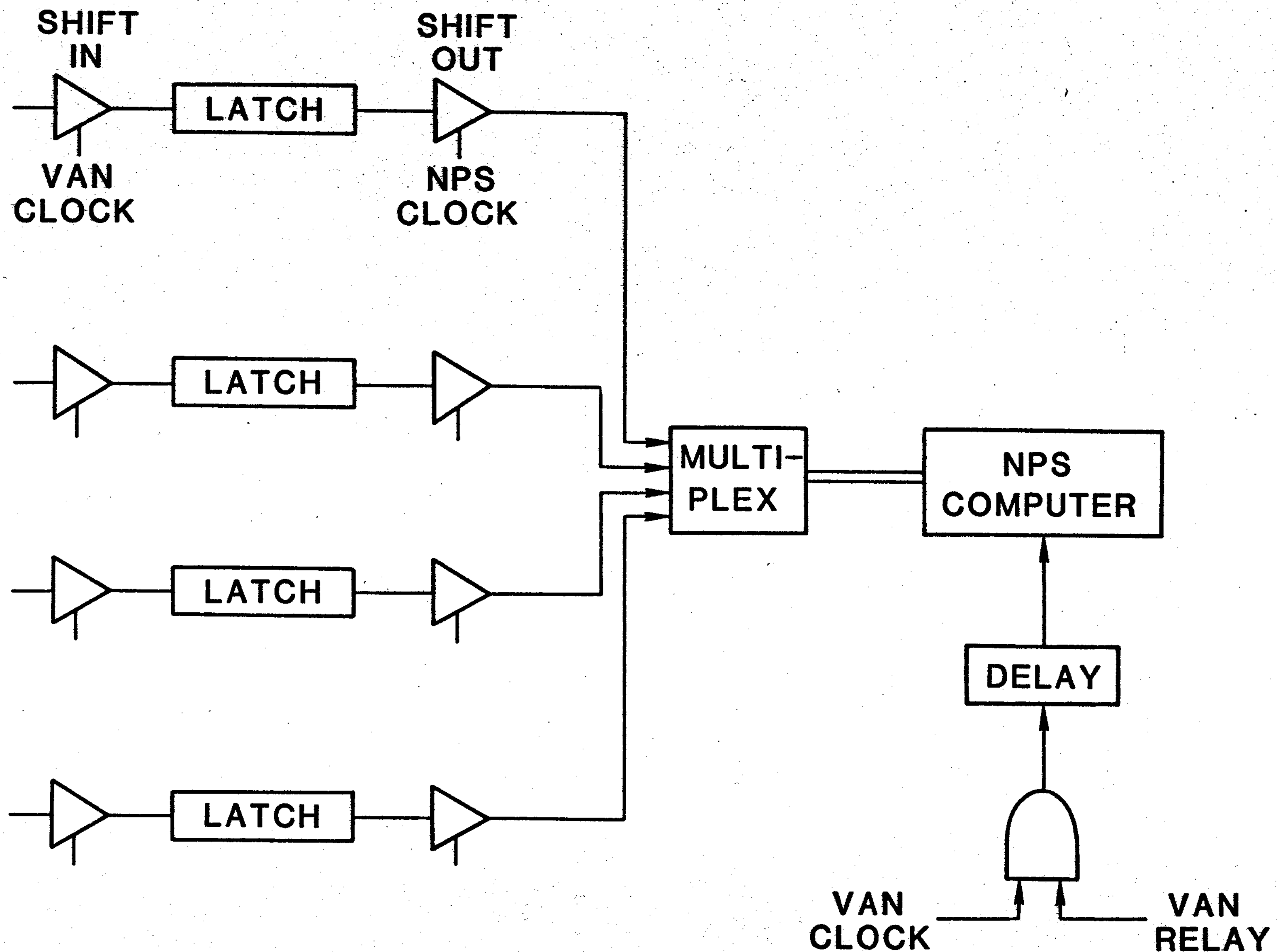


Figure 9. Basic Components of the NPS Data Acquisition System

Note on Figure 9 that the simultaneous presence of any relay and clock signals initiate the interrupt process. The command clock is used since it is only present when the command word is present. The interrupt doesn't occur until a pre-specified time delay has passed. This delay is needed to insure that all command and data words and the relay closure have been loaded into the latches. Without the delay, partial data could be transferred to the buss.

The timing of the command and data words and the NPS system are shown in Figure 10. A full cycle takes 22 msec. The computer interrupt is timed so that it occurs approximately 1 msec after the last data word is completely loaded into its shift register. The figure shows noise that is present on wire-lines 3 and 4 during the time between the end of the command word and the beginning of the data word. The data clock for these wire-lines is always activated so this noise would be strobed into the shift registers as data bits, and contaminate the data. In order to prevent this, the data clock signal is blocked from the system and only enabled during the time the data word is present, as shown at the bottom of the figure. Note that there is no reason to believe that noise is not also present with the data word. This eventuality is guarded against in the quality assurance processing to be described in the last section of this report.

A more detailed, but certainly not complete, block diagram of the equipment is shown in Figure 11. The figure shows the basic components that make up the input logic of the acquisition system for one of the wire-lines. If any relay is closed the system will

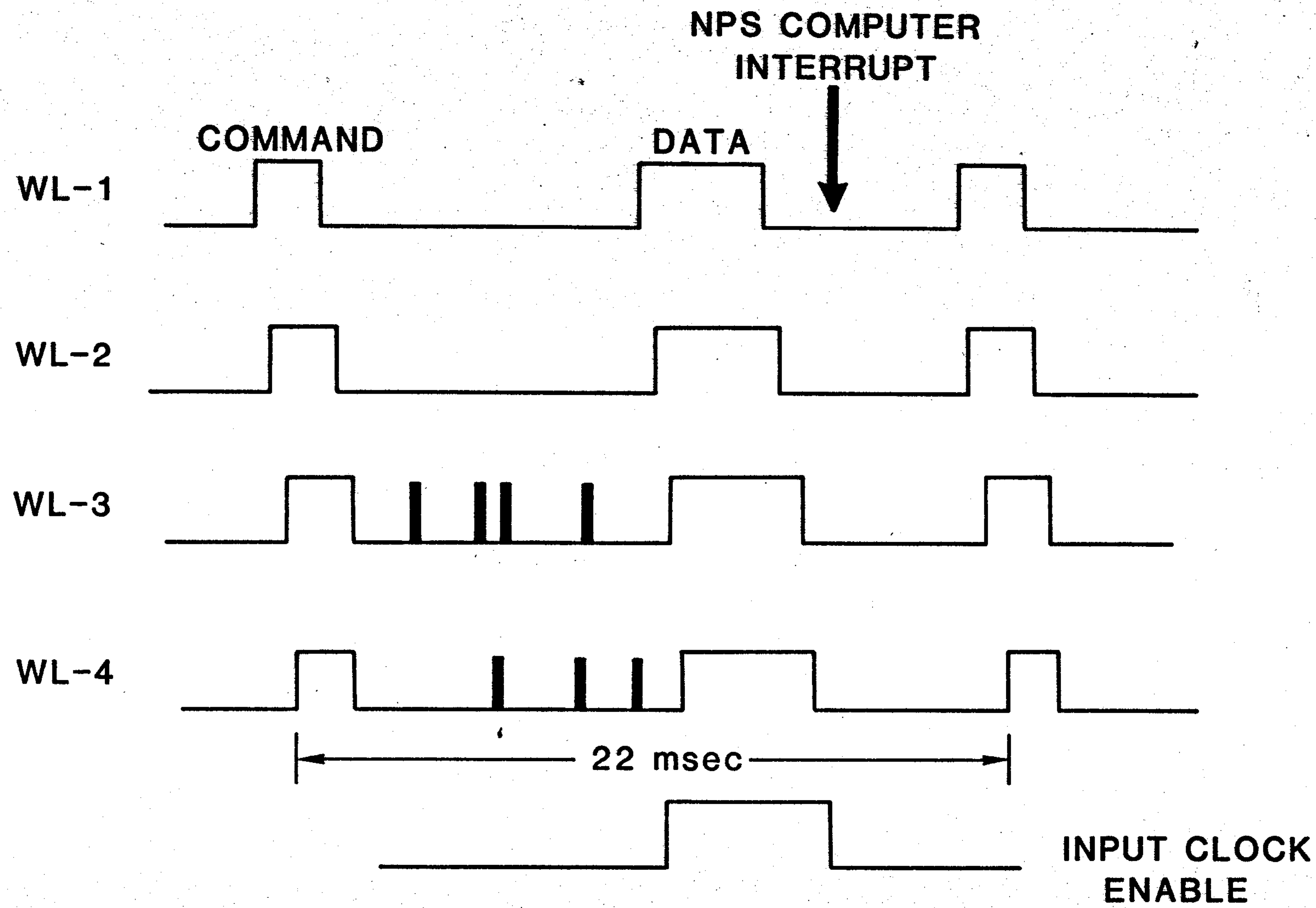


Figure 10. Time Line of the Command and Data Words and NPS Data Acquisition System Timing

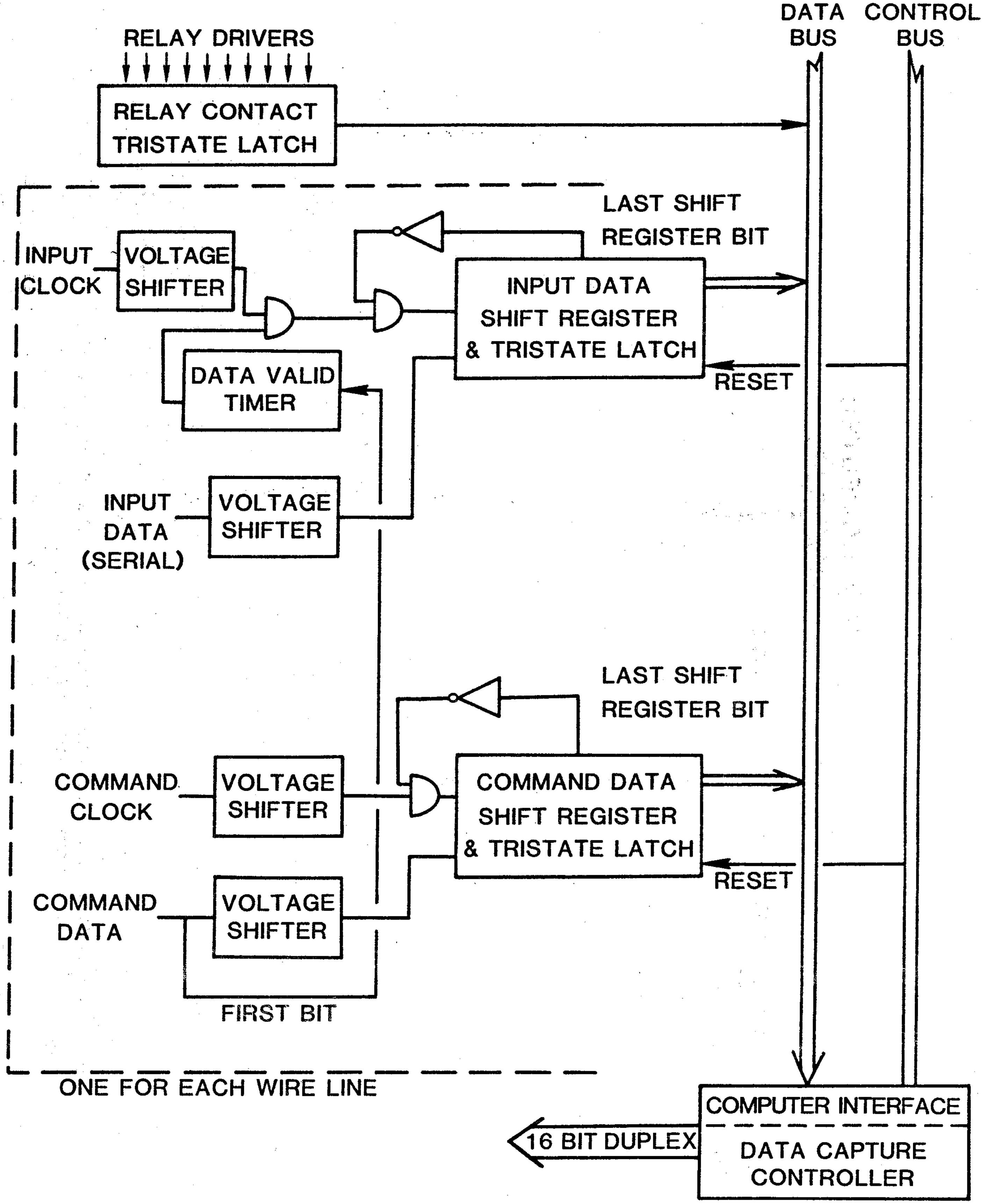


Figure 11. Block Diagram of the NPS Data Acquisition System

accept data. This is accomplished by the data capture controller activating the system through its reset function. Note that the Vandenberg system has the unfortunate characteristic that the last relay used on wire-lines 3 and 4 are left closed during the 0.2 sec dead time when there is no data. This can cause the generation of false interrupts on noise and use up valuable computer time. This problem is taken care of in software, as described in the next section.

Since the command clock is on only when the command word is present there is little problem with that data being contaminated. As soon as the shift register is filled, the last bit is sensed and the input to the register deactivated. The register is then ready for the command to shift data onto the buss.

The operation of the data shift register is much the same except for the need to deactivate the data clock for part of the cycle, as described above. This is done by using the first bit of the command word to start the "data valid timer", which activates the input when the pre-determined delay has elapsed. The delay is such that the data clock signal is enabled approximately 1 msec before the expected arrival of the data word.

All control logic for the system is hard-wired into the data capture controller. The controller and the computer handshake in order to insure that the computer is ready to accept data before it is multiplexed onto the buss. There is no need to go into the details of the controller functions in this document. Circuit diagrams of the system are included in Appendix B.

VII. DATA ACQUISITION METHODOLOGY

The 9826 computer is running flat-out to accomplish all of the needed data acquisition and processing tasks. Time saving methods for both acquisition and processing are an absolute necessity in order to accomplish this project without a large computer. The two principal time saving processes used are the pre-calculation of as many of the needed quantities as possible and the extensive use of integer precision arithmetic. An example of pre-calculation is the natural functions: all sines and cosines (2047 of each) are calculated at the initialization of the program and stored in an array. When needed they are obtained very rapidly using the array as a look-up table. All integers needed in the program are assigned an integer-precision variable name at program initialization.

VII-1 INPUT DATA STORING

The description of the NPS data acquisition system hardware in the former section shows how data and command words are acquired and multiplexed onto the 16-bit interface buss. The second critical part of the process is to correctly load the data into the computer memory in the very short time allowed. This must be done while operating on an interrupt basis during processing of data already stored. Two words arrive at the computer for each data acquisition, the command/relay word and the data word, the former being used to assign the proper memory location for the latter. The structure of the command/relay word is as follows:

	relay closed							command					
	6	5	4	3	2	1	N	*	*	*	*	*	*
bit	13	12	11	10	9	8	7	6	5	4	3	2	1

The relay closed is indicated by the bit value being 1 in the proper location. (The maximum number of relays on any wire-line that are used by the NPS system is 6.) Bit-7 of the word is unused, 6 bits are needed for the command word, yielding a total word length of 13 bits. The maximum decimal size of this word is 2 exp(12), which is 4096. The word is decoded to its decimal value and this number used to access a 4096 element look-up table. The numbers stored in the look-up table are the assigned locations in the input-data array for the input data. The codes assigned to the various sensors acquired are listed in Table 8. Obviously, since these are 94 sensors and 4098 elements in the look-up table, most of the numbers correspond to no sensor. The dummy numerical value of 95 is used for those numbers.

VII-2 ERROR TRAPPING

For reasons that are unknown, there are several types of errors that frequently occur in both the command and data words. The errors which have been observed are:

Command Code	009	014	052	054	055	056	101	102	103	200	300	299	301		Sensor Type
0	1		5	7	25	27	9	11	13	15	17		19		12' WD
1	2		6	8	26	28	10	12	14	16	18		20		12' WS
2			61	62	63	64	65	66	67	68	69		70		6' T
3			71	72	73		74	75	76	77	78		79		54' T
4			21	23			29	31	33	35	37		39		54' WD
5			22	24			30	32	34	36	38		40		54' WS
6												47			108' WD
7												48			108' WS
8		3													12' WD
9		4													12' WS
10								41		43	45		49		102' WD
11								42		44	46		50		102' WS
12										51	53		55		204' WD
13										52	54		56		204' WS
14													87		6' DP
15											80		81		102' T
16													92		Vis
17													93		Bar
18											57		59		300' WD
19											58		60		300' WS
20										82	83		84		204' T
21											85		86		300' T
22													88		54' DP
23													89		102' DP
24													90		204' DP
25													91		300' DP
26													94		SW DP

Table 8. Sensor Data Acquisition Codes

1. data words of 0,
2. simultaneous command and data words of 0,
3. data words of max value (2047),
4. sudden jumps in data word value.

There are several possible causes for these errors, some of which have been observed. Loss of synchronization is the main cause, which happens when one of the sites is returning intermittently poor data. (Loss of the air conditioning at site 200 has caused this condition.) The way in which the NPS system listens to the Vandenberg system and operates on an interrupt basis makes it susceptible to errors generated by timing jitter. The means used to capture these errors as the data are input to the computer are described below.

A zero command word corresponds to 12 ft wind direction. Thus an erroneous word would contaminate this data. As far as we have been able to determine, erroneous command word of 0 occurrences are always accompanied by a 0 data word so that the error can be trapped. Erroneous 0 data words that are accompanied by the correct command word also occur. Both types of errors are trapped by not accepting data with zero data words.

Unfortunately, zero wind speed and wind direction do occur and this test will eliminate those data values. This is not much of a problem since only a word value of true zero will be eliminated. The maximum word value is 2047, and one part in 2047 translates into 0.09 deg and 0.043 kts. Thus, wind speed and direction values less than these values will be eliminated by the test and there is a slight bias against low values in the program.

Maximum data word values occasionally occur erroneously on all wire-lines, and always occur on wire-lines 3 and 4 during the dead time when no real data is being transmitted. It is also possible for bit-value errors to cause data words that are abnormally large. Thus, there is a test for data word value that eliminates any word that is larger than a maximum value. For example, wind speeds greater than 70 kts or temperatures greater than 100 deg F are not expected to occur. The program does not accept any data word that is greater than 1700. The value of this maximum can be changed with season.

The most perplexing problem that occurs is the sudden change in data word value. Such changes can not be captured by either the zero or maximum test and are difficult to distinguish from changes that are due to the naturally occurring turbulence. Strangely enough, the most common error is an increase in the data word value by a factor of two from one one-second period to the next. This error is trapped by storing the last data entry for a sensor and comparing the new data entry to it. If the increase in the new value over the old is more than 70%, the new value is rejected and the old value remains the stored value. If this test had to be made for all sensors, the time required would be too long to complete data acquisition during one cycle. Fortuitously, this type of error only occurs for a few of the sensors, which were identified after some confusing results appeared, and the errors are possible to trap. No reasonable explanation for the occurrence of this error has been found.

VIII. DATA PROCESSING AND STORAGE

VIII-1 INPUT DATA PROCESSING

There is a synchronization problem caused by operating the computer on an interrupt basis. It is possible, even likely, that an interrupt will occur at a time in the processing cycle that will cause the data to be changed in the middle of a calculation or for the computer to attempt to process data that has not been updated. The required processing performance is that one full set of data be processed each data acquisition cycle and when this is complete, the computer begins on the next set of data. In order to insure this sequencing, a dual input array system is used. The input array is divided into buffer (BUF) and data (DAT) sections. One section accepts input data and the other is used for data processing and they are switched back and forth from input to processing when both steps are completed. A flow chart of the data acquisition process is shown in Figure 12.

After processing is complete, DAT is dummied. This is necessary because no data may be input for some of the sensors during the next cycle because they fail one of the data acquisition error checks. In that eventuality, the old data would be used again in the next processing cycle if DAT were not dummied.

In order for the BUF/DAT system to work, two crucial criteria must be met:

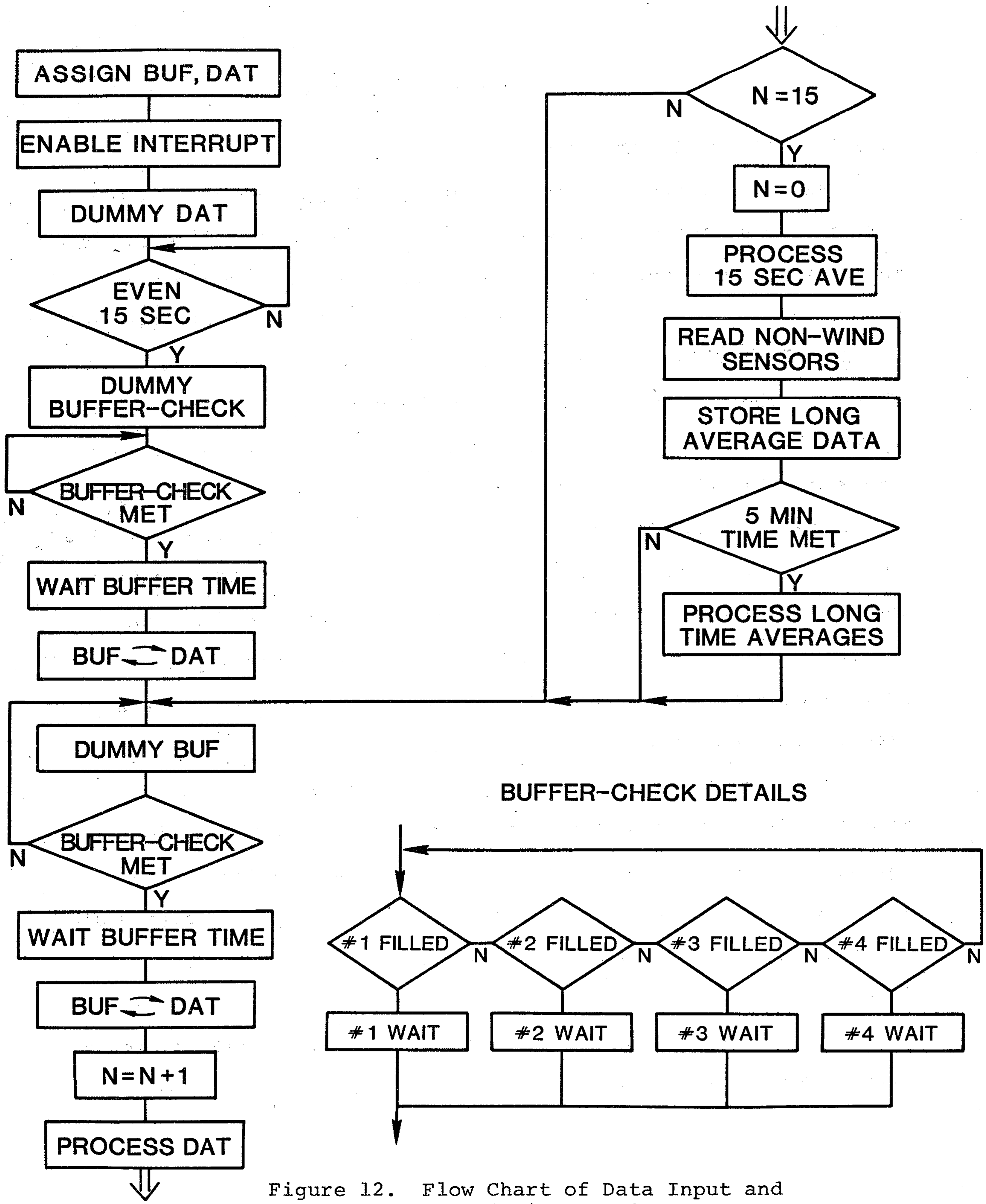


Figure 12. Flow Chart of Data Input and Calculation of Time Averages

1. BUF must be filled before it is switched to DAT,

2. processing DAT must take less time than the acquisition cycle.

If No. 2 were not met, the only adverse affect would be to mix data from two successive data cycles and to have a slight reduction in the amount of data obtained. The resulting averages would not be seriously affected, but some timing problems would be encountered.

If BUF is not filled before it is switched to DAT, data will be lost and a complete breakdown in synchronization can occur. To prevent this from happening, four sensors are assigned to be buffer-filled check sensors. These sensors are chosen to be near the end of the data acquisition cycle for the wire-line to which they are assigned. At the completion of each interrupt and data input, these four sensors are polled to see if data has been input to any one of them. If it has, a preset wait period is initiated to insure that all data is acquired. The wait time depends on the sensor that passed the data-inputted test since not all buffer-check sensors are at the same point in the sensor call-up sequence.

Those sensors used in the buffer-check are dummied with the value 2048 before data input to BUF begins. 2048 is not a possible data word so that any data input will change this value. Note that it is not correct to make the test on an empty array since 0 is a possible input data value.

Since sensors, towers, and complete wire-lines can be out of service, it is necessary to have the buffer-check sensors distributed so that equipment malfunction will not leave the system with no active sensors to check. The program has several possible sensors that can be used whose codes and wait times are stored in an array. The sensors to be used are accessed by their sensor codes and can be chosen at time of program initialization by entering their codes or by using four default sensors. Also, the sensors can be easily changed while the program is running as the need arises. The sensors available for use for buffer checks are listed in Table 9.

As has been mentioned above, one of the quality assurance checks that is made on the data is a comparison of the Vandenberg and NPS 15 min averages. In order for this check to be valid, it is necessary that both systems process the same data. Thus, the NPS system must be time synchronized to the Vandenberg real-time clock. Maintaining this synchronization is not easy in view of the way data is acquired. Time jitter in the Vandenberg system can cause cycles of data to be missed, and, if this occurs often enough, loss of synchronization can occur. The following is a description of the program logic that insures that synchronization is maintained:

1. Data acquisition is not begun until an even 15 sec time occurs (0, 15, 30, or 45 sec real-time).
2. A counter is used to determine when 15 sets of data have been processed, then a 15 sec average calculated. Note that this step can introduce some time drift.

Wire-line	Tower	Sensor	Code	Wait-time
1	103	dT54	76	0.308
		WD54	33	0.33
	009	WD12	1	0.44
		WS12	2	0.46
2	200	dT204	82	0.04
		WD204	51	0.11
	101	dT54	74	0.66
		WD54	59	0.64
3	300	dT300	85	0.264
		WD300	57	0.286
	299	WD108	47	0.594
		WS108	48	0.616
4	301	SW	94	0.044
		DP300	91	0.066
		dT300	86	0.088

Table 9. Available Buffer-check Sensors and Their Associated Wait Times

3. At the end of every data acquisition cycle (every 1 sec) the computers real-time clock is read to determine if the time is an even 5 min (5, 10, 15 min, etc.).
4. When an even 5 min occurs, a 5 min average is produced and the time is checked to determine if 15 min and 1 hr averages should also be calculated.
5. The 5 min check can cause an interruption of the acquisition process before 15 cycles have been completed. If this causes the number of acquisitions to be less than 10, the 15 sec average for that period is not calculated.

VIII-2 DATA MANAGEMENT AND STORAGE

At the end of each data acquisition cycle, all wind sensor data is processed to produce the vector components of the wind for each wind speed/wind direction pair. This information is then added into the appropriate 15 sec storage arrays for later processing to produce that average. The non-wind sensor data is not processed at the end of each cycle, it is lost. When a 15 sec average is produced, the non-wind data for the last acquisition cycle is added into the long-term average storage array. Thus, the frequency of acquisition of non-wind sensor data is a factor of 15 less than for the wind data. This is not a problem since it is only for the wind data that high frequency spectral information is desired.

When a 15 sec average is calculated, the stored 15 sec data is added into the long-term average storage array so that that array contains all wind data. The flow of the data through the arrays, from acquisition to final storage, is shown in Figure 13.

Note that the figure shows that some of the storage arrays are integer precision. This is done in order to save time in data processing, as was mentioned in the former section. The use of integer precision requires that the storage be in individual arrays rather than one large array since some stored quantities are too large and require full precision. In fact, the 15 sec average is the longest that can be used and still utilize integer precision since 15 maximum length data words sum to the maximum size allowed: (15 = 5 bits) x (2047 = 10 bits) = 15 bits.

The data processing is to produce means and standard deviations of the wind variables and to produce means for the non-wind sensors. Also, the final data stored should have those quantities included that allow calculation of means and standard deviations for any time period longer than 15 sec that is desired. This requires that the following parameters be stored:

1. sum of x-component,
2. sum of y-component,
3. sum of x-squared,
4. sum of y-squared,
5. sum of xy,
6. number of data points,

and for the non-wind sensors,

7. sum of sensor data,
8. number of data points.

When the appropriate time period has elapsed, the desired output quantities are calculated, scaling factors applied, and the data stored on tape (and disc for the 15 min averages). All storage is integer precision, which is done to save storage space. Full precision requires 8 byte words and integer 2 byte, a factor of 4 savings. With the use of integer precision, both storage media can hold 2 weeks of data. Scaling factors are needed so that data precision will not be lost when converting to integer precision.

The output data, their positions in the output arrays, and the conversion factors that have been applied, are shown in Table 10. Even though the data stored are different, the array sizes for the 15 sec and longer averages are the same. To have different sizes would require different record sizes on the output tape which would make using the data difficult. Using the same size arrays means that there is not enough space in the long-average arrays to hold all of the non-wind sensor counters, which poses no problems in using the data.

Note that one array number is not used in the longer average output arrays. This location is saved to be used for inversion height, which will be inserted at a later time.

The mean xy component, which is in the 15 sec output, will often be a number which is too large for integer precision. To prevent integer overflow, the square root of this number is used. The sign of the number is preserved and care should be taken not to lose the sign when decoding the number.

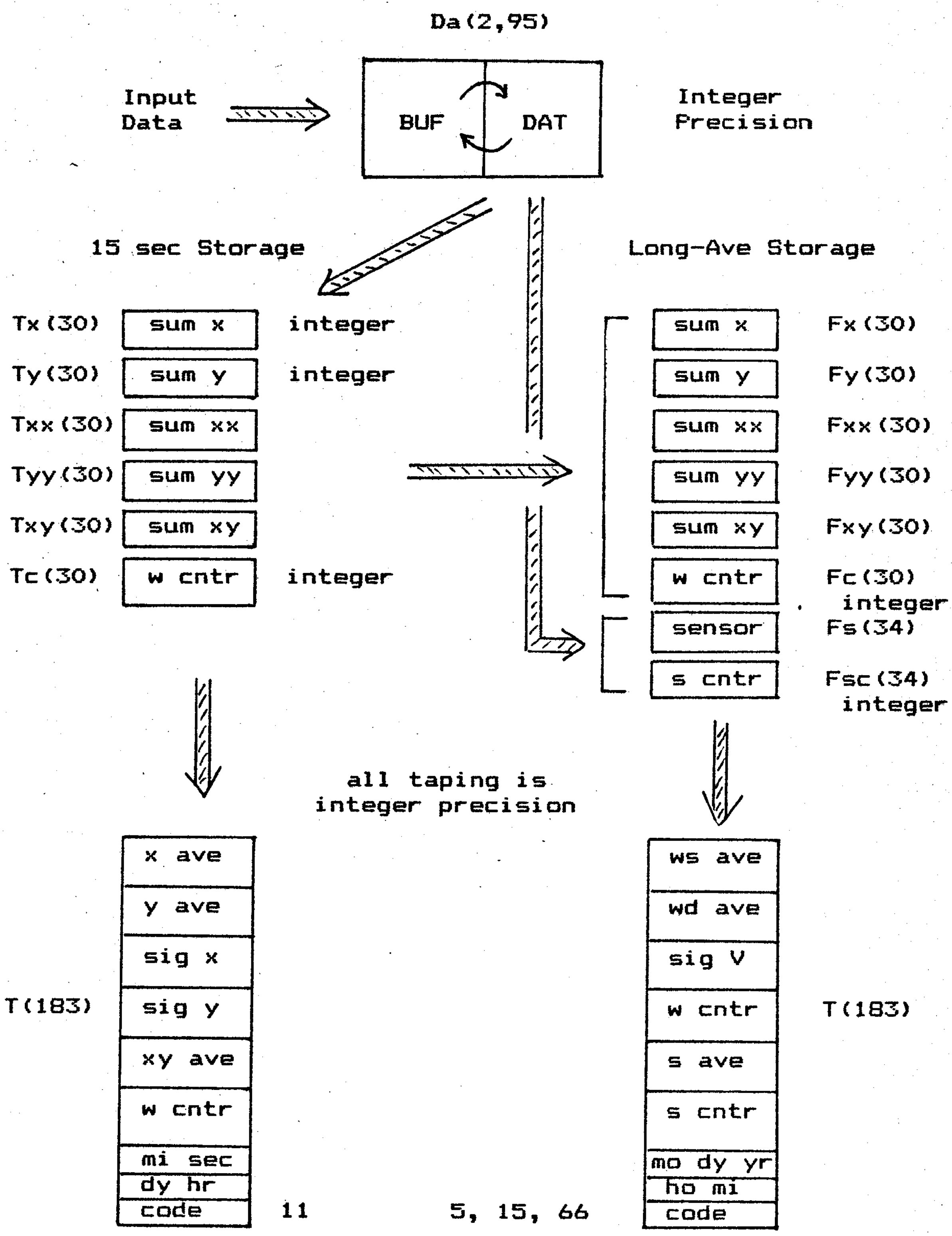


Figure 13. Flow of Data Through the Arrays from Acquisition to Storage

quantity output 15 sec average	output array positions	conversion factors
mean x-component	1-30	X10
mean y-component	31-60	X10
x standard deviation	61-90	X100
y standard deviation	91-120	X100
mean xy	121-150	10 * SQR
counter	151-180	X1
min, sec	181	100Xmin + sec
day, hour	182	100Xdy + hr
code = 11	183	X1

Longer Averages		
mean wind speed	1-30	X10
mean wind direction	31-60	X10
cross-wind standard deviation	61-90	X10
wind counter	91-120	X1
mean sensor value	120-154	T X100, other X10
sensor counter	155-179	X1
(empty)	180	
month, day, year	181	1000Xmo + 10Xdy + 4
our, minute	182	100Xho + min
ode = 55, 15, 66	183	X1

Table 10. Output Data and Conversion Factors

IX. QUALITY ASSURANCE

IX-1 ON SITE SYSTEM ANALYSIS

Rejection of obviously erroneous data is partly accomplished at the time of data acquisition, as was described in Section VII. That analysis is far from complete because only the most obvious errors can be trapped and because, due to processing time limitations, the checks could only be applied to some sensors. There are two more levels of quality control applied to the data:

1. comparison of NPS and Vandenberg 15 min averages,
2. analysis of all averages that are stored on 9-track tape.

The second method will be described in the second part of this section.

Comparison of the 15 sec averages is done on-site at Vandenberg in order to catch, as quickly as possible, problems with data acquisition. It was originally designed to find problems with the NPS system but analysis of the results show that problems with the Vandenberg systems are also detected. Recall that Figure 2 is a copy of a 15 sec average print-out.

The comparison is labor intensive since it requires number by number comparison of the complete print-out. As the performance of the system has improved, it has only been necessary to check once every 6 hours of data, four averages for each day. When problems are detected, more print-outs can be checked for the time when data is suspect to attempt to isolate the problem.

The most obvious error that is detected is when the results from the two systems do not agree. However, other types of errors are detected. If the data that is transmitted by the Vandenberg system is in error, meteorological values that are unreasonable will appear. For example, the wind direction reported for a sensor may be far different than any other for that area, or the wind speed for a lower level on a tower may read higher than that for an upper level. Such errors may be obvious when the print-outs are examined.

The main purpose of the on-site analysis is to enable problems with the system to be corrected as soon as possible. The second advantage is to identify, and record, those times when data were suspect. Such data can either be checked in later quality assurance analysis or removed from the data set.

IX-2 DATA ERROR ANALYSIS

The main quality assurance method employed is the extensive analysis of the averages contained on 9-track tapes. Note that this is not a check of the original data which was acquired, but an analysis of the subsequent averages. Any data that does not pass the quality assurance tests is assigned the value 999. New tapes are produced with the "bad" data removed in this manner. The original tapes are saved in case the analysis has been done unsatisfactorily or in case a different analysis procedure is needed.

One point should be emphasized: it is not possible to eliminate all errors that occur in the data acquisition procedure by the methods to be described here, or by any method one can conceive of. Errors that occurred when the individual data points were acquired are "smoothed over" when the data is averaged. The main effects of individual errors will be to increase the standard deviations, and if the errors are not random, to skew the averages. What one hopes to do with a good quality assurance procedure is to eliminate those data whose characteristics deviate substantially from the norm. If the analysis procedures are too restrictive, natural statistical fluctuations of the atmosphere can be lost and certainly interesting events will be lost.

The most challenging part of the quality assurance procedure is to decide where to draw the threshold between natural fluctation and error. We do not assume that we have been able to do so correctly with the decision criteria described in this section. Analysis of the results may expose the need for re-analysis of the data at a later date. In fact, the next section describes statistical analyses of the data that will allow us to determine how well the quality analysis procdure is working. Note that the procedures that are currently being used have been developed after extensive analysis of the data that has been acquired at Vandenberg and are probably as good as is currently available.

One caveat should be mentioned: the procedure does not test for certain types of peculiarities in the data, such as the lower level wind being higher than the upper or flow reversal with height. At Vandenberg such events can be real. We leave it to the user of the data to decide which tests of this type should be made.

The following is a description of the current quality assurance procedures. The tests that are made are:

1. sensor dummied-out,
2. out of valid limits,
3. value remains constant,
4. change from one average to the next (rate) too large.

The flow chart of the program that is used to make these tests is shown in Figure 15.

DUMMIED-OUT SENSORS

The first test listed requires that the sensor to be dummied-out be known before the program is run for a particular data set. This information is obtained from the on-site comparison of the 15 min averages and Vandenberg records. Any sensor which is giving consistently poor results for a period of time is dummied-out for that period so that data is not used. Such occurrences depend on the performance of NPS and Vandenberg hardware and it is not unusual for the data from some sensor to be eliminated in this manner.

OUT OF VALID LIMITS

The test to determine if the data is within valid limits is more stringent than that made during the time of data acquisition. The limits of validity are established after examination of the 15 min averages, which allows narrower limits to be set if no extreme meteorological conditions were encountered. Since this test is made on averaged data, its value can be questioned since out of

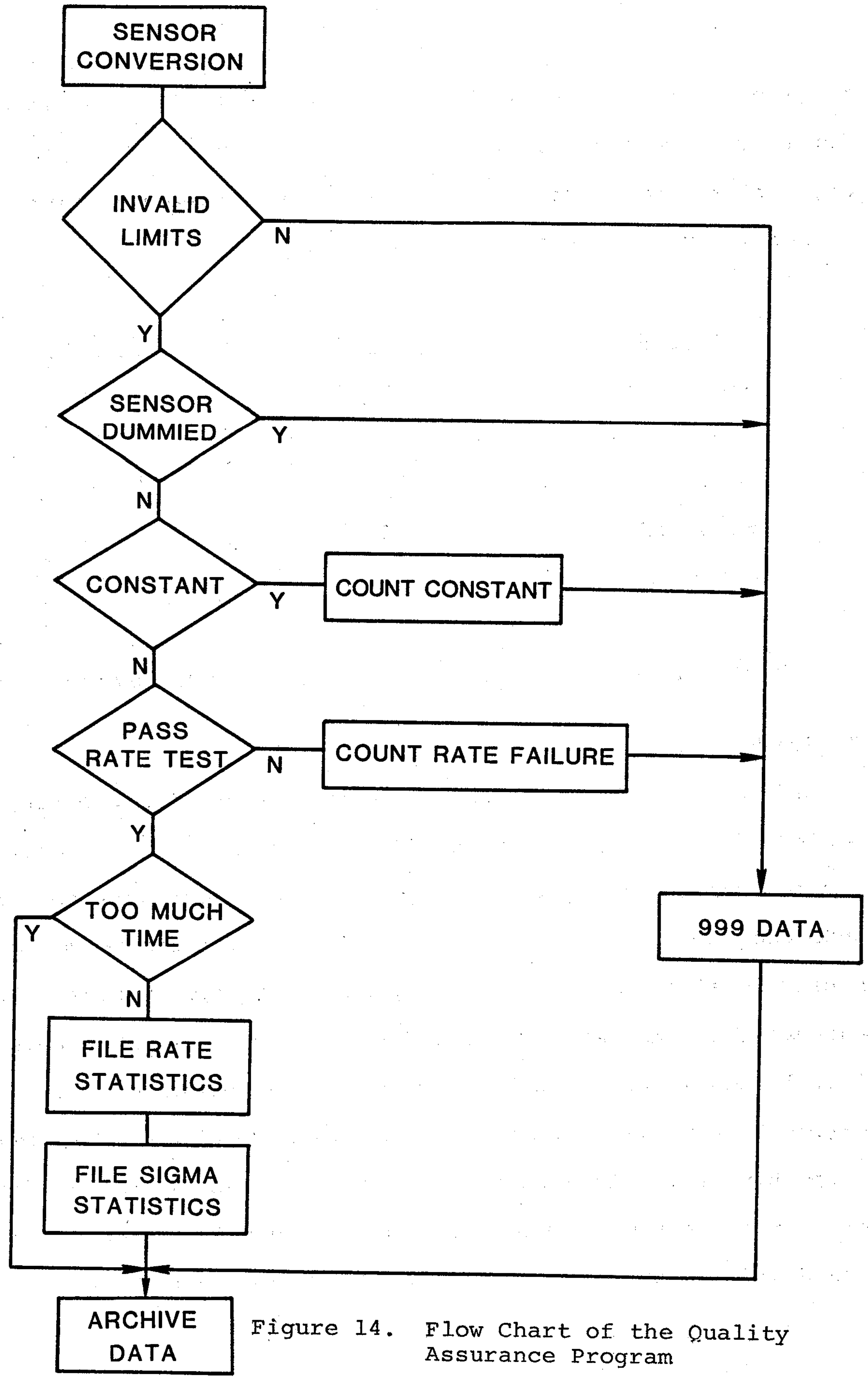


Figure 14. Flow Chart of the Quality Assurance Program

range data is averaged with other good data. The test is included for sake of completeness, and because the more restrictive limits may capture some errors. The maximum and minimum allowable limits currently in use are shown in Table 11.

Sensor	Minimum	Maximum
wind speed	0	60
wind direction	0	360
temperature	30	90
temp difference	-5	+5
dew point	10	90

Table 11. Data-Valid Limits for Quality Assurance Tests

CONSTANT VALUE

If the data from a sensor remains constant for too long a period of time, it is probable that the sensor is malfunctioning (or calibration being performed) since the atmosphere normally undergoes continual small changes. If the average value for a sensor remains constant for 5 successive 15 sec averages, it is assumed that the data is bad and all of the constant values 999'd. (For 5 min, 15 min, and 1 hr averages, the number of successive values that cause failure of the constant test are 5, 4, and 3, respectively). Bit errors can occur and an error in the least significant bit corresponds to 0.04 kts and 0.090 deg for wind speed and direction respectively. Thus, averages that are within 1.5 times these values of each other are assumed to have the same value for purposes of this test.

The temperature sensors are quite stable and have a long time constant. Thus, small fluctuations in the temperature measurements (air temperature and dew point temperature) are less likely to occur than for the wind sensors. Also examination of the data has shown that bit errors for these sensors rarely occur. Thus, the temperature averages are only assumed to be constant if the average values are identical.

RATE OF CHANGE

Meteorological parameters can only change so fast and if the reported rate of change is too large the data is suspect. Unfortunately, it is difficult to know just what constitutes "too large" a rate of change. Certainly, a wind speed change from 2 to 20 kts in 15 sec or a temperature change of 50 deg F in 5 min is not reasonable. A wind direction change of 90 deg in 15 sec is unreasonable if the wind speed is 20 kts but is not unreasonable if the speed is 1 kt. Establishing the boundaries between the expected rate of change and the erroneous value is very difficult.

Maximum accepted changes from one averaging period to the next have been established for the wind and temperature sensors. These values are listed in Table 12 for temperature and Table 13 for wind sensors. The rates have purposely been set large because we do not have sufficient knowledge of the naturally occurring fluctuations in the atmosphere within complex terrain to make unambiguous choices. As statistics are obtained from the quality assurance program, it will be possible to specify the acceptable rate limits more closely.

Note that the rate tests described here are essentially the same as those done on a few sensors during data acquisition. The main difference is that there the tests were done on the data acquired from one measurement cycle to the next, and these quality assurance tests are being done on the stored averages.

It is not possible to make the rate test if the elapsed time from one averaging period to the next is too long. This can occur if the system is shut down for a period of time or if an average is missing due to loss of synchronization. If the elapsed time is greater than the averaging period, but less than twice the averaging period, the rate test is made but with the maximum acceptable rate 50% larger than the normal value. If the elapsed time is more than twice the averaging period, the rate test is not made.

	Averaging Period		
	5 min	15 min	10 min
Statistics	4 deg F	6 deg F	8 deg F
Rate Fail	8 deg F	12 deg F	15 deg F

Table 12. Quality Assurance Maximum Acceptable Changes and Rate Statistics Maximum Values for Temperature

COMMENT

Examination of some of the data after quality assurance shows that some care should be exercised in using the data, which is not surprising. When a sensor or a portion of the system is malfunctioning, much of the resulting data will be 999'd but not necessarily all. Data that contains a significant number of 999s for a given period should probably not be used for modeling purposes. We leave it up to the user to decide what constitutes a "significant" number of 999's.

WIND CATEGORY	Wind Speed (kts)								Wind Direction (deg)							
	Statistics				Rate Fail				Statistics				Rate Fail			
	15s	5m	15m	1hr	15s	5m	15m	1hr	15s	5m	15m	1hr	15s	5m	15m	1hr
1	5	10	10	15	30	30	90	90	180	180	180	180	180	180	180	180
2	5	10	10	15	30	30	90	90	120	150	160	160	180	180	180	180
3	6	10	10	15	30	30	90	90	90	120	140	140	140	150	180	180
4	6	10	10	15	30	30	90	90	45	90	120	120	140	150	150	180
5	7	10	10	15	30	30	90	90	45	60	100	120	100	120	150	150
6	7	10	10	15	30	30	90	90	45	60	80	80	100	120	120	150
7	8	10	10	15	30	30	90	90	30	45	60	60	80	100	120	150
8	9	10	10	15	30	30	90	90	30	45	60	60	60	90	100	120
9	9	10	10	15	30	30	90	90	30	30	40	60	60	90	100	120
10	10	10	10	15	30	30	90	90	30	30	40	60	60	90	100	120

Table 13. Quality Assurance Maximum Acceptable Changes and Rates Statistics Maximum Values for Wind Sensors

X. STATISTICAL ANALYSIS

As has been described at length in the former section, not enough is known of the behavior of the atmosphere, and certainly not its behavior in complex terrain, to allow unambiguous choices to be made of the quality assurance test thresholds. Of course, it is of great interest to modelers to determine the expected scales or frequency distributions of the fluctuations in complex terrain. For these reasons, a complete statistical analysis of these data is being done.

The simplest statistics being gathered are the number of times the sensors fail the constant tests and the rate tests. This information will be useful for analyzing the behavior of the Vandenberg and NPS systems.

Extensive statistics on the rates of change of the various sensors for each of the averaging periods are being gathered. For each sensor and averaging period, the change in the average value from one averaging period to the next is binned in the following way:

1. A maximum rate for statistical analysis is established and 9 equal width bins below this value and one bin for all rates greater than this value are used.
2. 10 wind speed bins, with each bin being 3 kts wide, from 0 to 30 kts, with the last bin containing all speeds greater than 27 kts, are used.

The maximum rates for statistical analysis are listed in Tables 12 and 13.

This binning strategy yields 100 bins for each sensor-average category. The number of bins may seem excessive, but wind speed is obviously an important parameter and the rate should be binned fine enough to allow a reasonable analysis to be made. The real worry is that important parameters may be left out, such as wind direction, which would be useful for indicating oragraphic effects.

When the standard deviation for a parameter is large, one would expect that the change in the parameter from one averaging period to the next would be correspondingly large. In fact, one would expect that the ratio of the change to the standard deviation would be a parameter that could be used for quality assurance. Statistics on the ratio of the change to standard deviation for the former averaging period are being gathered. This analysis is called sigma statistics.

The sigma statistics binning is much the same as for the rate statistics. The wind speed bins are the same. The ratio is divided into 9 bins of unit width and a 10th bin for all ratios greater than 9. We anticipate that once the results of this analysis are well underway, a quality assurance sigma test will be developed.

No statistics are gathered for those cases where the elapsed time between the averages is greater than the averaging period.

XI. VECTOR AVERAGING

The horizontal wind vector is decomposed into vector components, with the x-direction defined to be E and the y-direction N. Angles are measured from N using the common convention of positive rotation being clockwise looking down. All processing and storage of wind data is done utilizing the vector components.

The main reason for using vector averaging is to be able to use the recorded data to produce averages over any time period desired. For monatonic variables, such as temperature and speed, it is a simple matter to form longer averages from shorter as long as the number of data points in each average is known. Wind direction is a problem because angles repeat every 360 deg. For example, if two successive averages are 60 deg and 310 deg, the calculated average of them will be 5 deg. However, the wind may have turned through 180 deg during the data acquisition time, in case the actual average direction was 185 deg. Without vector averaging, there is no way to distinguish between these two results.

The remainder of this section describes the use of the vector components to produce the desired quantities. The quantities are means and standard deviations of the vector components, the speed and direction, and the crosswind standard deviation.

For any measured parameter, P, the mean, $\overline{P}$, and standard deviation, σ_p, are defined to be:

$$\overline{P} = \sum_i P_i / N, \tag{1}$$

$$\sigma^2 = \sum_i (P_i - \overline{P})^2/(N - 1) \tag{2}$$

where N is the number of points that make up the average. In what follows, we will drop the subscripts i. The second equation reduces to the useful form:

$$\sigma^2 = (\Sigma P^2 - N\overline{P}^2)/(N - 1). \tag{3}$$

For the case where a longer-term average is to be determined from two shorter averages, the mean is

$$\overline{P} = (N_1 \overline{P}_1 + N_2\overline{P}_2)/(N_1 + N_2) \tag{4}$$

where the subscripts indicate the two shorter time periods. The standard deviation is given by

$$(N - 1)\sigma^2 = (N_1 - 1)\sigma_1{}^2 + (N_2 - 1)\sigma_2{}^2 + (\overline{P}_1 - \overline{P}_2)^2 N_1 N_2/N \tag{5}$$

where $N = N_1 + N_2$.

The relationships between the vector components and the speed and direction are needed for the analyses described earlier in this report. We use the symbols U, V, and θ to designate wind speed, cross-wind component, and wind direction, respectively. Of course, the average value of V is zero but its standard deviation is not. U and θ will be understood to mean the average values of those quantities.

By definition, the speed and direction are related to the vector components by

$$U = \bar{\bar{x}} + \bar{\bar{y}} , \qquad (6)$$

and

$$\tan\theta = \overline{y} / \overline{x} . \qquad (7)$$

The standard deviations are

$$\sigma_U{}^2 = \sigma_x{}^2 \sin^2\theta + \sigma^2 \cos^2\Theta + 2\sin\theta\cos\theta(\overline{xy} + \overline{x\,y}) N/(N - 1) \qquad (8)$$

and

$$\sigma_V{}^2 = \sigma_x{}^2 \cos^2\Theta + \sigma_y{}^2 \sin^2\Theta - 2\sin\Theta\cos\Theta(\overline{xy} - \overline{x}\,\overline{y}) N/(N - 1) \qquad (9)$$

The standard deviation of the wind direction is found from the cross-wind standard deviation by

$$\tan \sigma_\Theta = \sigma_V/U \qquad (10)$$

The rationale for the data processing and storage methods that have been used in this project become apparent when examining the above equations. The quantities needed to obtain the standard deviations are those that are stored in the data acquisition program, particularly the cross component xy which is not available from most data acquisition procedures.

APPENDIX A: COMPUTER PROGRAMS

Listed in this appendix are the computer programs used in acquiring and analyzing the Vandenberg meteorological data. A brief description of each of these programs is given below:

VANDAS	Data acquisition program
VDM	Acquire and print the binary data from the 16-bit interface buss for a single wire-line.
VDMPRINT	Acquire and print simultaneous decimal data from the 16-bit interface buss for all wire-lines.
DISCPRINT	Print data records from a floppy disc.
TAPEPRINT	Print data records from the 9-track tape.
QADISC	Print times, dates, and record numbers for records on the floppy discs.
VANQA	Print selected 15 min averages from the floppy discs for comparison with the Vandenberg 15 min averages.
QA	Quality assurance program for the 9-track tape data.

Listings of these programs make up the remainder of this appendix.

```
10!   RE-STORE "VANDAS"
20    !DATA ACQUISITION PROGRAM FOR VANDENBERG, TEST VERSION
30    !
40    OPTION BASE 1
50    INTEGER I,J,A,B,C,D,E,F,G,H,Ad,Ab,Buf,Dat,Su,Rs(2),On,Tw,Fo,Onf
60    INTEGER Thi,Six,Nin,Ont,Mo,Ho,Thr,Ten,R,Hun,Off,Nf,Z,Zer,Dumy
61    INTEGER Tofs,Iz,Iy,Thrt,Twn,Nn,Nnf,Nnh,Thh,Tsh,Tnn,Dd,Et,Enz,Nhu
63    INTEGER Fif,Tn,Fofo,Fofi,T,Cou,Tofe,So,Cha,Chb,Chc,Chd,Cqq
70    DIM Time$[12],Month$(12)[3],Mo$[3],Bufch(15,2),Si(2049),Co(2049)
80    DIM Txx(30),Tyy(30),Txy(30),Fx(30),Fy(30),Fxx(30),Fyy(30),Fxy(30)
90    DIM Fix(30),Fiy(30),Fixx(30),Fiyy(30),Fixy(30),Fis(34),Z$[1]
100   DIM Hrx(30),Hry(30),Hrxx(30),Hryy(30),Hrxy(30),Hrs(34),Fa(34)
110   DIM Tax(30),Tay(30),Taxx(30),Tayy(30),Taxy(30),Tas(34),Tac(30)
120   INTEGER Tx(30),Ty(30),Tc(30),Fc(30),Fic(30),Hrc(30),Fn,Elf,Nff
130   INTEGER Da(2,95),Tape(183),Fsc(34),Fisc(34),Hrsc(34),N,Fur,Fiv
131   INTEGER Tnt,Sev,Twl,Twfi,Thfo,Yy,Are,Tho,Big,Bigs,Tim,Old(3),Zz
140   INTEGER Assd(28,14),Relay(4,6),Lup(4,8192),Ig,Wl,Rl,Code,In(8)
150   DIM Hms$[8],Dmy$[11],Tim$(2)[12],Yr$[4],Tasc(34),P$[1]
160   !
170   ! ################### Setting Up the System ####################
171   !
172   Yr$="1984"
174   RESET 7
180   On=1
190   Zer=0
191   T=0
200   PRINTER IS 1
201   GOSUB Datime
202   PRINT USING "//"
203   PRINT "              CHECK TIME AND DATE "
204   PRINT " "
205   PRINT "               IF CORRECT   (CONT)"
206   PRINT " "
208   PRINT "               IF INCORRECT   (RUN)"
209   PAUSE
210   PRINT USING "24/"
211   P$="N"
212   INPUT "IS THIS A POWER FAILURE RESTART ? (Y) (CONT)",P$
213   IF P$="Y" THEN 221
214   PRINT "     install DATA DISCS    (CONT)"
215   PAUSE
216   Z$="N"
217   INPUT "Do you want to INITIALIZE DISCS ? (Y) (CONT)",Z$
218   IF Z$="Y" THEN GOSUB Discinit
219   GOTO 260
221   PRINT "       are DATA DISCS Installed ?"
230   PRINT USING "/"
240   PRINT "       if not INSTALL now (CONT)"
250   PAUSE
260     PRINT USING "24/"
270     PRINT "SETTING UP"
280     PRINT USING "4/"
290     GOSUB Setup
```

```
300     GOSUB Lookupt
310     PRINT USING "24/"
330   PRINT USING "4/"
360        IF P$<>"Y" THEN 490
370        PRINT "put DATA DISC in Right Hand Drive"
380        PRINT USING "3/"
390        PRINT "**** RESET TAPE RECORDER ****"
400        PRINT USING "/"
410        PRINT ".TIGHTEN TAPE --- POWER ON --- LOAD--RESET ---"
420        PRINT USING "/"
430        PRINT "           --- ON LINE ---(CONT)"
440        PAUSE
490     PRINT USING "24/"
500     OUTPUT 718;"BS(4026)"
510     GOSUB Bufcheck
550     PRINT USING "24/"
560     IF P$="Y" THEN 750
580     INPUT "RE-START Data Discs ?  (Y)",Z$
590     IF Z$<>"Y" THEN 660
600     OUTPUT @Fil3,1;Zer
610     OUTPUT @Fil4,1;Zer
620     R=On
660   Z$="N"
670   INPUT "DO YOU WANT TO RE-START TAPE ? (Y)",Z$
680        IF Z$<>"Y" THEN 750
690        Z$="N"
700        INPUT "ARE YOU SURE YOU WANT TO RE-START TAPE ? (Y)",Z$
710        IF Z$<>"Y" THEN 750
720        OUTPUT 718;"RW"
730   !##################################################################
740   !
750   ! ************** Create Integers and Zero Matracies ************
760   P$="N"
770   T=0                                          !Tape Counter
780   Su=6144                                      !Subtract from Inputs
790   Dumy=8192                                    !Dummy Data Value
800   Tofs=2045
810   Tofe=2048
811   Bigs=7844
813   Big=1400
814   Old(1)=Big
815   Old(2)=Big
816   Old(3)=Big
820   Rs(1)=4
830   Rs(2)=0
840   Dd=0
860   Tw=2
870   Thr=3
880   Fo=4
881   Fiv=5
884   Elf=11
885   Twl=12
886   Thrt=13
890   Fur=14
```

```
891   Fif=15
893   Twn=20
894   Twfi=25
895   Tn=29
900   Thi=30
901   Thfo=34
902   Fofo=44
903   Fofi=45
905   Ff=55
906   Fn=59
910   Six=60
911   Sev=7
912   Et=8
914   So=61
915   Nf=94
916   Nff=95
920   Nin=90
930   Ont=120
940   Onf=150
950   Ten=10
960   Hun=100
961   Tnt=293
962   Tnn=299
964   Thh=300
965   Enz=890
966   Nhu=900
970   Tho=1000
980   Tsh=3600
990   Off=154
1000  !
1010  MAT Tc= (0)
1020  MAT Tx= (0)
1030  MAT Ty= (0)
1040  MAT Txx= (0)
1050  MAT Tyy= (0)
1060  MAT Txy= (0)
1070  MAT Fc= (0)
1080  MAT Fx= (0)
1090  MAT Fy= (0)
1100  MAT Fxx= (0)
1110  MAT Fyy= (0)
1120  MAT Fxy= (0)
1130  MAT Fs= (0)
1140  MAT Fsc= (0)
1150  MAT Fic= (0)
1160  MAT Fix= (0)
1170  MAT Fiy= (0)
1180  MAT Fixx= (0)
1190  MAT Fiyy= (0)
1200  MAT Fixy= (0)
1210  MAT Fis= (0)
1220  MAT Fisc= (0)
1230  MAT Hrc= (0)
1240  MAT Hrx= (0)
```

```
1250  MAT Hry= (0)
1260  MAT Hrxx= (0)
1270  MAT Hryy= (0)
1280  MAT Hrxy= (0)
1290  MAT Hrs= (0)
1300  MAT Hrsc= (0)
1310  !
1320  !****** BEGINNING OF DATA ACQUISITION **************************
1330  Ab=1                                              !Set Input Buffer
1340  Ad=-1
1350  Buf=(Ab+3)/2
1360  Dat=(Ad+3)/2
1370  !
1380  ENABLE INTR Ca;1
1390 Dated:                                   !Set Up Time/Date for Storage
1400           Time$=FNTime$(TIMEDATE)
1410           Date$=FNDate$(TIMEDATE)
1411           Dy=VAL(Date$[1,2])
1412           Yr=VAL(Date$[11,11])
1413           Mo$=Date$[4,6]
1414           FOR Mo=On TO Twl
1415             IF Mo$=Month$(Mo) THEN 1418
1416           NEXT Mo
1418           Daytap=Yr+Ten*Dy+Tho*Mo
1419           PRINT Daytap
1421 !
1430 Start: !  <-<-<-<- START IS HERE <-<-<-<-<-<-<-<-<-<-<-<-
1431        N=Zer
1432        Nn=Zer
1433        Nnf=Zer
1434        Nnh=Zer
1440  MAT Tape= (Zer)
1450  MAT Da= (Dumy)                                !Dummy Input Matrix
1460       I=Zer                             !Check 15sec Time Condition
1470       Sod=TIMEDATE MOD 86400
1480       I=I+On
1490   DISP "WAITING for even 15sec    #SEC =";INT((Sod MOD 60)*10)/10
1500       IF I<10000 THEN 1530
1510       PRINT "15SEC TIME CONDITION NOT MET"
1520       GOTO Start
1530       IF Sod MOD 15<14.1 THEN 1470
1540  Da(Buf,Cha)=Dumy                             !Dummy Input Buffer
1550  Da(Buf,Chb)=Dumy
1560  Da(Buf,Chc)=Dumy
1570  Da(Buf,Chd)=Dumy
1580       I=Zer
1590       IF Da(Buf,Cha)<Dumy THEN 1680            !Test Input Buffer
1600       IF Da(Buf,Chb)<Dumy THEN 1700
1610       IF Da(Buf,Chc)<Dumy THEN 1720
1620       IF Da(Buf,Chd)<Dumy THEN 1740
1630       I=I+On
1640       DISP "waiting for FILLED BUFFER"
1650       IF I<Dumy THEN 1590
1660       DISP "BUFFER NOT FILLING"
```

```
1670       GOTO Start
1680  Twait=Chat
1690  GOTO 1750
1700  Twait=Chbt
1710  GOTO 1750
1720  Twait=Chct
1730  GOTO 1750
1740  Twait=Chdt
1750  Start=TIMEDATE
1760  IF TIMEDATE-Start<Twait THEN 1760
1770  Ab=-Ab
1780  Ad=-Ad
1790  Buf=(Ab+Thr)/Tw
1800  Dat=(Ad+Thr)/Tw
1810  FOR I=On TO Nff
1820       Da(Dat,I)=Dumy
1830  NEXT I
1840  !
1850  !####################### Data Transfer Loop ###################
1860 Loop: !
1870       I=On
1880       IF Da(Buf,Cha)<Dumy THEN 1960             !Test Input Buffer
1890       IF Da(Buf,Chb)<Dumy THEN 1980
1900       IF Da(Buf,Chc)<Dumy THEN 2000
1910       IF Da(Buf,Chd)<Dumy THEN 2020
1920       I=I+On
1930       IF I<Dumy THEN 1880
1940       PRINT "BUFFER NOT FILLING"
1950       GOTO Start
1960  Twait=Chat
1970  GOTO 2030
1980  Twait=Chbt
1990  GOTO 2030
2000  Twait=Chct
2010  GOTO 2030
2020  Twait=Chdt
2030  Start=TIMEDATE
2040  IF TIMEDATE-Start<Twait THEN 2040
2050  Ad=-Ad
2060  Ab=-Ab
2070  Buf=(Ab+Thr)/Tw
2080  Dat=(Ad+Thr)/Tw
2090  N=N+On
2091       DISP N
2092    Zz=6                                 !WS-12 SPECIAL TESTS
2093    Yy=1
2095    GOSUB Checkws
2098    Zz=12
2099    Yy=2
2100    GOSUB Checkws
2102    Yy=3
2103    Zz=20
2105    GOSUB Checkws
2106    GOTO 2118
```

```
2107 Checkws:    !
2108             Qqq=Da(Dat,Zz)-Su
2109    IF Qqq=Zer OR Qqq>Big THEN 2116
2110    IF Qqq>1.6*Old(Yy) THEN 2112
2111    GOTO 2114
2112    Qqq=Qqq/Tw
2113    Da(Dat,Zz)=Qqq
2114    Old(Yy)=Qqq
2116    RETURN
2118  FOR I=On TO Fn STEP 2                        !Transfer Data From Buffer
2120       A=I+On
2130       B=Da(Dat,I)-Su+On
2131       IF B=Tofe OR B<Tw THEN 2250
2140       Z=Da(Dat,A)-Su
2150       IF Z>Big OR Z<On THEN 2250
2160       C=A/Tw
2170       Tc(C)=Tc(C)+On
2180       X=Z*Si(B)
2190       Y=Z*Co(B)
2200       Tx(C)=X+Tx(C)
2210       Ty(C)=Y+Ty(C)
2220       Txx(C)=X*X+Txx(C)
2230       Tyy(C)=Y*Y+Tyy(C)
2240       Txy(C)=X*Y+Txy(C)
2250  NEXT I
2251  Sod=TIMEDATE MOD 86400
2252  Ttt=Sod MOD Thh
2253  IF Ttt>Tnn OR Ttt<On THEN 2320
2260  IF N>Fur THEN 2330                              !Check For 15 Samples
2270       FOR I=On TO Nf                                !Dummy Data Buffer
2280           Da(Dat,I)=Dumy
2290       NEXT I
2300     GOTO Loop
2310  !#################################################################
2311  !
2320  Nn=On
2330  Mi=Sod MOD Tsh DIV Six
2331  Ho=Sod DIV Tsh
2332  IF N<Ten THEN 2740
2340  Tape(183)=Elf
2350  Se=Sod MOD Six
2362  Tape(182)=100*Dy+Ho
2370  Tape(181)=Hun*Mi+Se
2380  PRINT Mi;"min";Se;"sec","# =";Tc(5)
2390  N=Zer
2400  !
2410  FOR I=On TO Thi                                 !Calculate Output
2411       E=I+Onf
2420       IF Tc(I)<Tw THEN 2570
2430       A=I+Thi
2440       B=I+Six
2450       C=I+Nin
2460       D=I+Ont
2480       F=Tc(I)-On
```

```
2490        Tape(I)=Tx(I)/Tc(I)                   !Tape is the Output Matrix
2500        Tape(A)=Ty(I)/Tc(I)
2510        Sigx=(Txx(I)-Tx(I)*(Tx(I)/Tc(I)))/F
2520        Sigy=(Tyy(I)-Ty(I)*(Ty(I)/Tc(I)))/F
2530        Tape(B)=Ten*SQR(ABS(Sigx))
2540        Tape(C)=Ten*SQR(ABS(Sigy))
2550        Tape(D)=SQR(ABS(Txy(I)))
2560        IF Txy(I)<Zer THEN Tape(D)=-Tape(D)
2570        Tape(E)=Tc(I)
2580    NEXT I
2590    OUTPUT @Tapd;Tape(*)                          !Tape 15sec Data
2591    T=T+On
2592    IF T<Elf THEN 2600
2593    T=Zer
2594    OUTPUT 718;"BS(4026)"
2600    MAT Fc= Tc+Fc                                 !15sec to 5min Storage
2610    MAT Fx= Tx+Fx
2620    MAT Fy= Ty+Fy
2630    MAT Fxx= Txx+Fxx
2640    MAT Fyy= Tyy+Fyy
2650    MAT Fxy= Txy+Fxy
2660    MAT Tc= (Zer)                                 !Empty 15sec Storage
2670    MAT Tx= (Zer)
2680    MAT Ty= (Zer)
2690    MAT Txx= (Zer)
2700    MAT Tyy= (Zer)
2710    MAT Txy= (Zer)
2720    MAT Tape= (Zer)
2740        FOR I=So TO Nf                            !Transfer Sensor Data
2750           IF Da(Dat,I)>Bigs THEN 2790
2751           IF Da(Dat,I)<On THEN 2790
2760           A=I-Six
2770           Fs(A)=Da(Dat,I)+Fs(A)-Su
2780           Fsc(A)=Fsc(A)+On
2790        NEXT I
2810    !                                             !Check For 5min Elapsed
2811    IF Nn=On THEN Five
2820    GOTO 2270
2830    !
2840 Five:  !
2860    Tape(183)=Ff                                  !Calculate 5min Output
2870    Tape(182)=Hun*Ho+Mi
2880    Tape(181)=Daytap
2890        PRINT Ho;":";Mi;":";INT(Se),"# in 5min =";Fc(5)
2900    MAT Tax= Fx
2910    MAT Tay= Fy
2920    MAT Taxx= Fxx
2930    MAT Tayy= Fyy
2940    MAT Taxy= Fxy
2950    MAT Tas= Fs
2960    MAT Tac= Fc
2970    MAT Tasc= Fsc
2980    GOSUB Tape                                    !Tape 5min Data
2990    OUTPUT @Tapd;Tape(*)
```

```
2991  Nn=Zer
2992  T=T+On
2993  IF T<Elf THEN 3020
2994  T=Zer
3000  OUTPUT 718;"BS(4026)"
3010  !
3020  MAT Fix= Fix+Fx                               !Transfer 5min to 15min
3030  MAT Fiy= Fiy+Fy
3040  MAT Fixx= Fixx+Fxx
3050  MAT Fiyy= Fiyy+Fyy
3060  MAT Fixy= Fixy+Fxy
3070  MAT Fic= Fic+Fc
3080  MAT Fis= Fis+Fs
3090  MAT Fisc= Fisc+Fsc
3100  MAT Fc= (Zer)                                    !Empty 5min Storage
3110  MAT Fx= (Zer)
3120  MAT Fy= (Zer)
3130  MAT Fxx= (Zer)
3140  MAT Fyy= (Zer)
3150  MAT Fxy= (Zer)
3160  MAT Fs= (Zer)
3170  MAT Fsc= (Zer)
3180  MAT Tape= (Zer)
3190  !
3191  IF Mi=Zer OR Mi=Fn THEN Disc
3200  IF Mi=Fur OR Mi=Fif THEN Disc             !Check for 15min Elapsed
3201  IF Mi=Tn OR Mi=Thi THEN Disc
3202  IF Mi=Fofo OR Mi=Fofi THEN Disc
3210  GOTO 2270
3220 Disc:                                          !Calculate 15min Output
3230        PRINT "on disc"
3240  MAT Tax= Fix
3250  MAT Tay= Fiy
3260  MAT Taxx= Fixx
3270  MAT Tayy= Fiyy
3280  MAT Taxy= Fixy
3290  MAT Tac= Fic
3300  MAT Tas= Fis
3310  MAT Tasc= Fisc
3320  Tape(183)=15
3321  Tape(182)=Hun*Ho+Mi
3322  Tape(181)=Daytap
3330  GOSUB Tape
3331  OUTPUT @Tapd;Tape(*)
3340  IF R>700 THEN 3380                       !Output 15min Data to Disc
3350           OUTPUT @Fil1,R;Tape(*)
3360           OUTPUT @Fil3,1;R
3370           GOTO 3400
3380       OUTPUT @Fil2,R-700;Tape(*)
3390       OUTPUT @Fil4,1;R-700
3400       R=R+On
3401    T=T+On
3403    IF T<Elf THEN 3410
3404    T=Zer
```

```
3405     OUTPUT 718;"BS(4026)"
3410   MAT Hrx= Fix+Hrx                      !Transfer 15min to 1hr Storage
3420   MAT Hry= Fiy+Hry
3430   MAT Hrxx= Fixx+Hrxx
3440   MAT Hryy= Fiyy+Hryy
3450   MAT Hrxy= Fixy+Hrxy
3460   MAT Hrc= Fic+Hrc
3470   MAT Hrs= Fis+Hrs
3480   MAT Hrsc= Fisc+Hrsc
3490   MAT Fic= (Zer)                              !Empty 15min Storage
3500   MAT Fix= (Zer)
3510   MAT Fiy= (Zer)
3520   MAT Fixx= (Zer)
3530   MAT Fiyy= (Zer)
3540   MAT Fixy= (Zer)
3550   MAT Fis= (Zer)
3560   MAT Fisc= (Zer)
3570   MAT Tape= (Zer)
3580   !
3590   IF Mi=Zer OR Mi=59 THEN Hour                !Check for 1hr Elapsed
3600   GOTO 2270
3610 Hour:                                          !Calculate 1hr Output
3620         PRINT Hrc(5);"in 1hr"
3630   MAT Tax= Hrx
3640   MAT Tay= Hry
3650   MAT Taxx= Hrxx
3660   MAT Tayy= Hryy
3670   MAT Taxy= Hrxy
3680   MAT Tas= Hrs
3690   MAT Tac= Hrc
3700   MAT Tasc= Hrsc
3710   Tape(183)=66                                  !Tape 1hr Data
3711   Tape(182)=Hun*Ho+Mi
3712   Tape(181)=Daytap
3720   GOSUB Tape
3730   OUTPUT @Tapd;Tape(*)
3740   T=T+On
3750   IF T<Elf THEN 3760
3751   T=Zer
3752   OUTPUT 718;"BS(4026)"
3760   MAT Hrc= (Zer)                               !Empty 1hr Storage
3770   MAT Hrx= (Zer)
3780   MAT Hry= (Zer)
3790   MAT Hrxx= (Zer)
3800   MAT Hryy= (Zer)
3810   MAT Hrxy= (Zer)
3820   MAT Hrs= (Zer)
3830   MAT Hrsc= (Zer)
3831   MAT Tape= (Zer)
3840   !
3845   IF Ho=23 AND Mi=59 THEN 3849                  !EOF for End of Day
3846   IF Ho=0 AND Mi=Zer THEN 3849
3847   GOTO 2270
3849   OUTPUT 718;"PB WE"
```

```
3850  GOTO Dated
3900  !
3910  !********Long Time Average Output Subroutine ******************
3920 Tape: !
3930  FOR I=On TO Thi                          !Calculations for Long Term
3940       Tape(I+Nin)=Tac(I)                        !   Average Output
3950       A=I+Thi
3960       IF Tac(I)<On THEN 4070
3970       Xs=Tax(I)*Tax(I)
3980       Ys=Tay(I)*Tay(I)
3990       Xsys=SQR(Xs+Ys)
4000       Tape(I)=Xsys/Tac(I)
4010       IF Tay(I)=Zer THEN 4050
4020       Bb=ATN(Tax(I)/Tay(I))
4021       Cc=10*((360+Bb) MOD 360)
4030       IF Tay(I)>Zer THEN Tape(A)=Cc
4040       IF Tay(I)<Zer THEN Tape(A)=10*(180+Bb)
4050       IF Tac(I)<Tw OR Xsys=Zer THEN 4070
4051    Bb=(Tayy(I)*Xs+Taxx(I)*Ys-2*Tax(I)*Tay(I)*Taxy(I))/(Tac(I)-On)
4060       Tape(I+Six)=SQR(Bb)/Xsys
4070  NEXT I
4080  FOR I=On TO Thfo
4090       IF Tasc(I)<On THEN 4110
4100       Tape(I+Ont)=Tas(I)/Tasc(I)
4110  NEXT I
4120  FOR I=On TO Twfi
4130       Tape(I+Off)=Tasc(I)
4140  NEXT I
4141  MAT Tax= (Zer)
4142  MAT Tay= (Zer)
4143  MAT Taxx= (Zer)
4144  MAT Tayy= (Zer)
4145  MAT Taxy= (Zer)
4146  MAT Tac= (Zer)
4147  MAT Tas= (Zer)
4148  MAT Tasc= (Zer)
4150  RETURN
4160  !
4161  !************** Subroutines for SETTING UP System **************
4170 Setup:  !
4180     Ca=12                             ! UART INTERFACE SELECT CODE
4190     ON INTR Ca,15 GOSUB Datgrab
4200     ASSIGN @It TO Ca;WORD,FORMAT OFF
4210  DISP "SETTING UP DISC STORAGE"
4220  ASSIGN @Fil1 TO "VANDAT:HP82901,700,0";FORMAT OFF
4230  ASSIGN @Fil2 TO "VANDAT:HP82901,700,1";FORMAT OFF
4240  ASSIGN @Fil3 TO "COUNTER:HP82901,700,0";FORMAT OFF
4250  ASSIGN @Fil4 TO "COUNTER:HP82901,700,1";FORMAT OFF
4260  ASSIGN @Tapd TO 719;FORMAT OFF
4270  !
4280  DISP "SETTING UP SIN & COS"
4281       DEG
4282       Si(1)=0
4283       Co(1)=1
```

```
4290  FOR I=1 TO 2047 STEP 2                              !Set Up sin & cos
4300       Aa=-.087890625*(I-1)
4301       Bb=.087890625*(I+1)
4310       Si(I+1)=SIN(Aa)
4320       Co(I+1)=COS(Aa)
4330       Si(I+2)=SIN(Bb)
4340       Co(I+2)=COS(Bb)
4350  NEXT I
4360  !
4370 DISP "CREATING LABELS"                               !Set Up Keys
4380  ON KEY 5 LABEL "STOP",15 GOTO Again
4390  ON KEY 8 LABEL "BUFCHECK",15 GOTO Buffer
4420  !
4425  DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
4450 DATA 76,.31,33,.33,1,.44,2,.46,82,.04,51,.11,74,.66,29,.64
4460 DATA 85,.262,57,.286,47,.594,48,.616,94,.044,91,.066,86,.088
4470 READ Bufch(*)
4480  !
4490 Discread:                                   !Read Record # Filed on Disc
4500                 DISP "READING DISC COUNTER"
4510       ENTER @Fil3,1;R
4520       IF R<700 THEN 4580
4530       ENTER @Fil4,1;R
4540       IF R=0 THEN 4550
4541       R=R+700
4542       GOTO 4580
4550       R=700
4580       R=R+1
4590       RETURN
4591  !
4600 Datime:                                     !DATE/TIME Setting Subroutines
4620  READ Month$(*)
4671    ENTER 716;Time$
4672    Mo=VAL(Time$[3,4])
4674    Mo$=Month$(Mo)
4675    Dmy$=Time$[5,6]&" "&Mo$&" "&Yr$
4676    Hms$=Time$[7,8]&":"&Time$[9,10]&":"&Time$[11,12]
4680!          OFF ERROR
4690! Dmy$=Time$[5,6]&" "&Mo$&" "&"1983"
4700! Hms$=Time$[7,8]&":"&Time$[9,10]&":"&Time$[11,12]
4710  SET TIMEDATE FNDate(Dmy$)+FNTime(Hms$)
4720  PRINT FNDate$(TIMEDATE)
4730  SET TIME FNTime(Hms$)
4740  PRINT FNTime$(TIMEDATE)
4750  RETURN
4760  !
4761  !###################### Utility Subroutines ###################
4770 Again:                                          !K5 STOP/RE-START Routine
4780        OUTPUT 718;"PB"
4790        Z$="N"
4800        INPUT "do you want END OF FILE on tape  (Y)",Z$
4810        IF Z$<>"Y" THEN 4880
4820        OUTPUT 718;"WE"
4830        Z$="N"
```

```
4840       INPUT "is this the END OF TAPE  (Y)",Z$
4850       IF Z$<>"Y" THEN 4890
4860       OUTPUT 718;"WE RW"
4870       GOTO 4920
4880       Z$="N"
4890       INPUT "REWIND TAPE ??  (Y)",Z$
4900       IF Z$="Y" THEN OUTPUT 718;"RW"
4910       Z$="N"
4920       INPUT "RESET DISCS TO START ??  (Y)",Z$
4930       IF Z$<>"Y" THEN 4960
4940       GOSUB Discread
4950       GOTO 4970
4960       R=1
4970       Z$="N"
4980       INPUT "INITIALIZE DISCS ??  (Y)",Z$
4990       IF Z$="Y" THEN GOSUB Discinit
5000       PRINT "YOU ARE READY TO RESTART"
5010       PRINT " "
5020       PRINT "EXECUTE  (CONT START)"
5030       PAUSE
5040       GOTO 6190
5050  !
5060 Discinit:                  !******** DISC INITIALIZATION *********
5070  PRINT "This Routine Initializes Discs for VANDENBERG"
5080  PRINT " "
5090  PRINT "it will WIPE OUT discs in the dual disc drive"
5100  PRINT " "
5110  PRINT "put NEW DISCS in the dual disc drive  (cont)"
5120  PAUSE
5130  Z$="N"
5131  PRINT "******* DON'T WIPE OUT DATA *******"
5140  INPUT "Ready to go Ahead ??  (Y) (CONT)",Z$
5150  IF Z$<>"Y" THEN 5260
5160  INITIALIZE ":HP82901,700,0"
5170  CREATE BDAT "COUNTER:HP82901,700,0",1,2
5180  CREATE BDAT "VANDAT:HP82901,700,0",701,366
5185  ASSIGN @Fil3 TO "COUNTER:HP82901,700,0";FORMAT OFF
5190  OUTPUT @Fil3,1;Zer
5200  INITIALIZE ":HP82901,700,1"
5210  CREATE BDAT "COUNTER:HP82901,700,1",1,2
5220  CREATE BDAT "VANDAT:HP82901,700,1",701,366
5225  ASSIGN @Fil4 TO "COUNTER:HP82901,700,1";FORMAT OFF
5230  OUTPUT @Fil4,1;Zer
5240  PRINT " "
5250  PRINT "   ** THE DEED IS DONE **"
5260  RETURN
5270  !
5280 Bufcheck:                             !Set Up Buffer Check Sensors
5290       Cha=85
5300       Chb=82
5310       Chc=94
5320       Chd=47
5330       PRINT "BUFFER CHECK SENSORS ARE";Cha;Chb;Chc;Chd
5340       IF P$="Y" THEN 5390
```

```
5350        Z$="N"
5360        INPUT "CHANGE BUFFER CHECK SENSORS ?  (Y)",Z$
5370        IF Z$<>"Y" THEN GOTO 5390
5380        INPUT "ENTER 4 SENSOR CODES A,B,C,D (CONT)",Cha,Chb,Chc,Chd
5390        FOR J=1 TO 15
5400          IF Cha=Bufch(J,1) THEN Chat=Bufch(J,2)
5410          IF Chb=Bufch(J,1) THEN Chbt=Bufch(J,2)
5420          IF Chc=Bufch(J,1) THEN Chct=Bufch(J,2)
5430          IF Chd=Bufch(J,1) THEN Chdt=Bufch(J,2)
5440        NEXT J
5450   RETURN
5460   !
5470 Buffer:                                             !Reset Buffer
5480   P$="N"
5481   GOSUB Bufcheck
5482   GOTO Start
5491   !
5500   !***** INTERUPT SERVICE ROUTINE & LOOKUP TABLE INITIALIZATION **
5510 Datgrab:         !
5520        ENTER @It;In(*)
5521!              IF In(Sev)=Su AND In(Et)=Zer THEN 5528
5522!              IF Dd=Zer THEN 5530
5523!              IF In(Et)=555 THEN 5530
5524!              In(Sev)=Dumy
5525!              Dd=Zer
5526!              GOTO 5530
5528!              Dd=On
5530        FOR Iz=On TO Fo
5540           Iy=Iz*Tw
5550           Da(Buf,Lup(Iz,In(Iy)+On))=In(Iy-On)
5560        NEXT Iz
5561        IF In(8)=364 THEN 5565
5562        IF In(4)=4176 THEN 5565
5563        GOTO 5570
5565        ON DELAY .16,14 GOSUB Enble
5566        GOTO 5580
5567 Enble:    !
5570        ENABLE INTR Ca;On
5580        RETURN
5590   !
5600   !************** Generate Lookup Table LUP(4,2048) **************
5610 Lookupt:             !
5620        Ca=12  !   UART INTERFACE SELECT CODE
5630        ON INTR Ca,15 GOSUB Datgrab
5640        ASSIGN @It TO Ca;WORD,FORMAT OFF
5650                       DISP "CREATING LOOKUP TABLES"
5660                       !    FIRST ENTER DEVICE ARRAYS
5670               !  1  2  3  4  5  6  7  8  9 10 11 12 13 14
5680    DATA          1,95, 5, 7,25,27,09,11,13,15,17,95,19,95,! 1
5690    DATA          2,95,06,08,26,28,10,12,14,16,18,95,20,95,! 2
5700    DATA         95,95,61,62,63,64,65,66,67,68,69,95,70,95,! 3
5710    DATA         95,95,71,72,73,95,74,75,76,77,78,95,79,95,! 4
5720    DATA         95,95,21,23,95,95,29,31,33,35,37,95,39,95,! 5
5730    DATA         95,95,22,24,95,95,30,32,34,36,38,95,40,95,! 6
```

```
5740   DATA          95,95,95,95,95,95,95,95,95,95,47,95,95,95,! 7
5750   DATA          95,95,95,95,95,95,95,95,95,95,48,95,95,95,! 8
5760   DATA          95,95,95,95,95,95,03,95,95,95,95,95,95,95,! 9
5770   DATA          95,95,95,95,95,95,04,95,95,95,95,95,95,95,! 10
5780   DATA          95,95,95,95,95,95,95,41,95,43,45,95,49,95,! 11
5790   DATA          95,95,95,95,95,95,95,42,95,44,46,95,50,95,! 12
5800   DATA          95,95,95,95,95,95,95,95,95,51,53,95,55,95,! 13
5810   DATA          95,95,95,95,95,95,95,95,95,52,54,95,56,95,! 14
5820   DATA          95,95,95,95,95,95,95,95,95,95,95,95,87,95,! 15
5830   DATA          95,95,95,95,95,95,95,95,95,95,80,95,81,95,! 16
5840   DATA          95,95,95,95,95,95,95,95,95,95,95,95,92,95,! 17
5850   DATA          95,95,95,95,95,95,95,95,95,95,95,95,93,95,! 18
5860   DATA          95,95,95,95,95,95,95,95,95,95,57,95,59,95,! 19
5870   DATA          95,95,95,95,95,95,95,95,95,95,58,95,60,95,! 20
5880   DATA          95,95,95,95,95,95,95,95,95,82,83,95,84,95,! 21
5890   DATA          95,95,95,95,95,95,95,95,95,95,85,95,86,95,! 22
5900   DATA          95,95,95,95,95,95,95,95,95,95,95,95,88,95,! 23
5910   DATA          95,95,95,95,95,95,95,95,95,95,95,95,89,95,! 24
5920   DATA          95,95,95,95,95,95,95,95,95,95,95,95,90,95,! 25
5930   DATA          95,95,95,95,95,95,95,95,95,95,95,95,91,95,! 26
5940   DATA          95,95,95,95,95,95,95,95,95,95,95,95,95,95,! 27
5950   DATA          95,95,95,95,95,95,95,95,95,95,95,95,94,95,! 28
5960         READ Assd(*)
5970         !
5980         !    NOW THE RELAY MATRIX
5990         !
6000         !       1  2  3  4  5  6
6010  DATA           1, 9,14,14,14,14,! 1
6020  DATA          14, 4, 5, 6, 7,10,! 2
6030  DATA           8,11,14,14,14,14,! 3
6040  DATA           3,13,14,14,14,14,! 4
6050          READ Relay(*)
6060          !
6070          !    NOW GENERATE LOOKUP TABLE LUP
6080           MAT Lup= (95)
6090          FOR Wl=1 TO 4       !  4 WIRELINES
6100            FOR Rl=1 TO 6     !  6 RELAYS
6110            Ig=2^(Rl+6)
6120              FOR Code=1 TO 28
6130              Lup(Wl,SHIFT(Code-1,-2)+1+Ig)=Assd(Code,Relay(Wl,Rl))
6140              NEXT Code
6150            NEXT Rl
6160          NEXT Wl
6161          DISP " "
6170          RETURN
6180  !
6190                           END
6200  !
6210  !************** DEFINED FUNCTIONS FOR DATE/TIME ***************
6220  DEF FNTime$(Now)  ! Given ``SECONDS' Return ``hh:mm:ss'
6230    !
6240    Now=INT(Now) MOD 86400
6250    H=Now DIV 3600
6260    M=Now MOD 3600 DIV 60
```

```
6270    S=Now MOD 60
6280    OUTPUT T$ USING "#,ZZ,K";H,":",M,":",S
6290    RETURN T$
6300  FNEND
6310  !
6320  DEF FNTime(T$)  !Given ''hh:mm:ss' Return ''SECONDS'
6330    !
6340    ON ERROR GOTO Err
6350    ENTER T$;H,M,S
6360    RETURN (3600*H+60*M+S) MOD 86400
6370 Err:OFF ERROR
6380      RETURN TIMEDATE MOD 86400
6390  FNEND
6400   DEF FNDate$(Seconds)   !Given ''SECONDS' Return 'dd mmm yyyy'
6410      !
6420      DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
6430      DIM Month$(1:12)[3]
6440      READ Month$(*)
6450      !
6460      Julian=Seconds DIV 86400-1721119
6470      Year=(4*Julian-1) DIV 146097
6480      Julian=(4*Julian-1) MOD 146097
6490      Day=Julian DIV 4
6500      Julian=(4*Day+3) DIV 1461
6510      Day=(4*Day+3) MOD 1461
6520      Day=(Day+4) DIV 4
6530      Month=(5*Day-3) DIV 153       ! MONTH
6540      Day=(5*Day-3) MOD 153
6550      Day=(Day+5) DIV 5             ! DAY
6560      Year=100*Year+Julian          ! YEAR
6570      IF Month<10 THEN
6580        Month=Month+3
6590      ELSE
6600        Month=Month-9
6610        Year=Year+1
6620      END IF
6630      OUTPUT D$ USING "#,ZZ,X,3A,X,4Z";Day,Month$(Month),Year
6640      RETURN D$
6650  FNEND
6660  !
6670  DEF FNDate(Dmy$)  !Given ''dd mmm yyyy' Return 'SECONDS'
6680    !
6690    DATA JAN,FEB,MAR,APR,MAY,JUN,JUL,AUG,SEP,OCT,NOV,DEC
6700    DIM Month$(1:12)[3]
6710    READ Month$(*)
6720    !
6730    ON ERROR GOTO Err
6740    I$=Dmy$&"     "
6750    ENTER I$ USING "DD,4A,5D";Day,M$,Year
6760    IF Year<100 THEN Year=Year+1900
6770    FOR I=1 TO 12
6780    IF POS(M$,Month$(I)) THEN Month=I
6790    NEXT I
6800    IF Month=0 THEN Err
```

```
6810    IF Month>2 THEN
6820      Month=Month-3
6830    ELSE
6840      Month=Month+9
6850      Year=Year-1
6860    END IF
6870    Century=Year DIV 100
6880    Remainder=Year MOD 100
6890    Julian=146097*Century DIV 4+1461*Remainder DIV 4
6891    Julian=Julian+(153*Month+2) DIV 5+Day+1721119
6900    Julian=Julian*86400
6910    IF Julian<2.08662912E+11 OR Julian>=2.143252224E+11 THEN Err
6920    RETURN Julian !Return Julian date in SECONDS
6930 Err:OFF ERROR          !ERROR in input.
6940      RETURN TIMEDATE     !Return current date.
6950  FNEND
```

```
10 !  RE-STORE  "VDM"
20    !   TEST VANDENBURG INTERFACE
30    OPTION BASE 1
40    INTEGER Stor(5000,2),In(8),Rs(2)
50     DIM T(200)
60        W11=1
70        Hpib=7
80        Ca=12
90        ASSIGN @It TO Ca;WORD,FORMAT OFF
100       Rs(1)=4    ! RESET LATCHES
110       Rs(2)=0
120       ON INTR Ca,15 GOSUB Service
130       PRINTER IS 1
150 Startup:              !
160                       Totc=0
170                       INPUT "ENTER NUMBER FRAMES",Ns
180                       INPUT "ENTER WL  NUMBER ",Wl
190                       W11=Wl*2-1
200        ENABLE INTR Ca;1
210           DISP "LOADING DATA "
220 Loop:               !
230                       DISP Totc
240                     IF Totc<=Ns THEN Loop
250 Out:            !
260                           OFF INTR Ca
270         PRINTER IS 701
280         St=1
290         En=Ns
300         PRINT "WIRELINE ";Wl
310         FOR It=St TO En
320           IF Stor(It,1)=-1 THEN Nx
330           PRINT Stor(It,1);TAB(10);BINAND(Stor(It,1),2047);TAB(18);
340             FOR Ii=0 TO 15
350             PRINT VAL$(BIT(Stor(It,1),15-Ii));
360             NEXT Ii
380             Dd=BINAND(Stor(It,2)/4,31)
390           PRINT TAB(38);Stor(It,2);TAB(46);Dd;TAB(50);
400             FOR Ii=0 TO 15
410               PRINT VAL$(BIT(Stor(It,2),15-Ii));
420             NEXT Ii
430             PRINT It
440 Nx:       NEXT It
450           !
470           PRINT USING "3/"
540                   PRINTER IS 1
550                   GOTO Ennd
560 Service:                     !
570            Totc=Totc+1
580            ENTER @It;In(*)
590            IF Totc>Ns THEN 700
600              FOR I=1 TO 2
610               Stor(Totc,1)=In(W11)
620               Stor(Totc,2)=In(1+W11)
```

```
630                 NEXT I
710                OUTPUT Ca USING "#,W";Rs(*)
720                      ENABLE INTR Ca;1
730                 RETURN
870 Ennd:                          END
```

```
10!    RE-STORE "VDMPRINT"
20     !LOAD VDM DATA IN COMPUTER AND PRINT
30     !
40     OPTION BASE 1
50     INTEGER I,J,Ad,Ab,Buf,Dat,Su,Rs(2),Gg,Iz,Iy,Ig,Wl,Rl,On,Tw,Fo
60     INTEGER Assd(28,14),Relay(4,6),Lup(4,8192),In(8),Ta(5000,8)
70     INTEGER Rel(4),Sens(4),Com(4),Da(2,95)
80     ! ######
90     On=1
100    Tw=2
110    Fo=4
120    Su=6144
130    PRINTER IS 1
140    PRINT USING "24/"
150      PRINT "SETTING UP"
160      PRINT USING "4/"
170      GOSUB Setup
190      PRINT USING "24/"
200    !
210    Ab=1
230    Buf=(Ab+3)/2
250    !
260 Start:    !
300    MAT Ta= (0)
310    MAT In= (0)
320    MAT Sens= (0)
330    MAT Rel= (0)
340    MAT Com= (0)
350    GOTO Loop
360    !
370 Setup:                                              !Getting System Ready
380       Ca=12  !  UART INTERFACE SELECT CODE
390       ON INTR Ca,15 GOSUB Datgrab
400       ASSIGN @It TO Ca;WORD,FORMAT OFF
430   RETURN
440    !
620    !         ** INTERUPT SERVICE ROUTINE
630 Datgrab:              !
640          ENTER @It;In(*)
690          Gg=Gg+1
700          ENABLE INTR Ca;On
710          RETURN
1290   !
1300 Loop:     INPUT "ENTER NUMBER LINES ?",Ns
1310           DISP "LOADING DATA "
1320           ENABLE INTR Ca;1
1330 !
1340 Gg=1                                               !Enter Data
1350       IF Gg>Ns THEN 1371
1351       Ta(Gg,1)=In(1)
1352       Ta(Gg,2)=In(2)
1353       Ta(Gg,3)=In(3)
1354       Ta(Gg,4)=In(4)
```

```
1355        Ta(Gg,5)=In(5)
1356        Ta(Gg,6)=In(6)
1357        Ta(Gg,7)=In(7)
1358        Ta(Gg,8)=In(8)
1360        GOTO 1350
1370 !
1371 DISABLE INTR Ca
1380          DISP "PRINTING DATA"
1430 Prints:        PRINTER IS 701
1440 PRINT TAB(4);"WIRE LINE 2";TAB(34);"WIRE LINE 3";TAB(64);"WIRE LINE 4"
1450          FOR J=1 TO 3
1460             PRINT TAB(30*(J-1));
1470             GOSUB Header
1480          NEXT J
1490          GOTO 1520
1500 Header:PRINT "SENSOR   REL   COM   ";
1510          RETURN
1520 FOR I=1 TO Ns
1530            DISP "MAX # ";Ns,"       PRINTING # ";I
1540        Sens(1)=Ta(I,1)-Su                        !Calculate Printed Data
1550        Sens(2)=Ta(I,3)-Su
1560        Sens(3)=Ta(I,5)-Su
1570        Sens(4)=Ta(I,7)-Su
1580        Rely=SHIFT(Ta(I,2),6)
1590           IF Rely=0 THEN 1610
1600           Rel(1)=LOG(Rely)/LOG(2)
1610        Rely=SHIFT(Ta(I,4),6)
1620           IF Rely=0 THEN 1640
1630           Rel(2)=LOG(Rely)/LOG(2)
1640        Rely=SHIFT(Ta(I,6),6)
1650           IF Rely=0 THEN 1670
1660           Rel(3)=LOG(Rely)/LOG(2)
1670        Rely=SHIFT(Ta(I,8),6)
1680           IF Rely=0 THEN 1700
1690           Rel(4)=LOG(Rely)/LOG(2)
1700        Com(1)=SHIFT(BINAND(Ta(I,2),124),2)
1710        Com(2)=SHIFT(BINAND(Ta(I,4),124),2)
1720        Com(3)=SHIFT(BINAND(Ta(I,6),124),2)
1730        Com(4)=SHIFT(BINAND(Ta(I,8),124),2)
1740 !
1750    PRINT Sens(2);TAB(10);Rel(2);TAB(15);Com(2);TAB(30);
1760    PRINT Sens(3);TAB(40);Rel(3);TAB(45);Com(3);TAB(60);
1770    PRINT Sens(4);TAB(70);Rel(4);TAB(75);Com(4)
1780    MAT Sens= (0)
1790    MAT Rel= (0)
1800    MAT Com= (0)
1810    NEXT I
1820 !
1830    MAT Ta= (0)
1840    PRINT USING "3/"
1850 !
1860 PRINTER IS 1
1870 END
```

```
10 !  RE-STORE  "VDM"
20    !   TEST VANDENBURG INTERFACE
30    OPTION BASE 1
40    INTEGER Stor(5000,2),In(8),Rs(2)
50     DIM T(200)
60         W11=1
70         Hpib=7
80         Ca=12
90         ASSIGN @It TO Ca;WORD,FORMAT OFF
100        Rs(1)=4   ! RESET LATCHES
110        Rs(2)=0
120        ON INTR Ca,15 GOSUB Service
130        PRINTER IS 1
140 !      GOTO 6011
150 Startup:              !
160                       Totc=0
170                       INPUT "ENTER NUMBER FRAMES",Ns
180                       INPUT "ENTER WL  NUMBER ",Wl
190                       Wll=Wl*2-1
200          ENABLE INTR Ca;1
210             DISP "LOADING DATA "
220 Loop:               !
230                       DISP Totc
240                     IF Totc<=Ns THEN Loop
250 Out:            !
260                           OFF INTR Ca
270           PRINTER IS 701
280           St=1
290           En=Ns
300           PRINT "WIRELINE ";Wl
310           FOR It=St TO En
320             IF Stor(It,1)=-1 THEN Nx
330             PRINT Stor(It,1);TAB(10);BINAND(Stor(It,1),2047);TAB(18);
340               FOR Ii=0 TO 15
350               PRINT VAL$(BIT(Stor(It,1),15-Ii));
360               NEXT Ii
380               Dd=BINAND(Stor(It,2)/4,31)
390             PRINT TAB(38);Stor(It,2);TAB(46);Dd;TAB(50);
400               FOR Ii=0 TO 15
410                 PRINT VAL$(BIT(Stor(It,2),15-Ii));
420               NEXT Ii
430               PRINT It
440 Nx:         NEXT It
450             !
470             PRINT USING "3/"
530 Stp:              !
540                     PRINTER IS 1
550                     STOP
560 Service:                    !
570              Totc=Totc+1
580              ENTER @It;In(*)
590              IF Totc>Ns THEN 700
600                FOR I=1 TO 2
```

```
610                  Stor(Totc,1)=In(W11)
620                  Stor(Totc,2)=In(1+W11)
630                  NEXT I
700     !  NEW
710                 OUTPUT Ca USING "#,W";Rs(*)
720                       ENABLE INTR Ca;1
730                  RETURN
740                  STOP
750 What:          !
760                STOP
780           OUTPUT @It;Rs(*)
790                GOTO 770
800                FOR I=0 TO 3
810                 FOR Ii=0 TO 7
820                 G=I*8+Ii
830                 PRINT G,I,Ii
840                 NEXT Ii
850               NEXT I
860               STOP
870                END
```

```
10!  RE-STORE "TAPEPRINT"
20   !ACCESS AND PRINT VANDENBERG TAPE FILES
30   !
40 OPTION BASE 1
50   INTEGER T(183)
60   OUTPUT 718;"BS(4026)"
61   Z$="N"
62   INPUT "REWIND TAPE ?? (Y)",Z$
63   IF Z$<>"Y" THEN 80
70   OUTPUT 718;"RW"
80   ASSIGN @Tapd TO 719;FORMAT OFF
90   PRINTER IS 701
100  FOR I=1 TO 35
110  ENTER @Tapd;T(*)
120  IMAGE 6D,6D,6D,6D,6D,6D,6D,6D,6D,6D
130  FOR K=1 TO 18
140  J=10*(K-1)
150  PRINT USING 120;T(1+J),T(2+J),T(3+J),T(4+J),T(5+J),T(6+J),T(7+J),
                    T(8+J),T(9+J),T(10+J)
160  NEXT K
170  PRINT USING "7D,7D,7D";T(181),T(182),T(183)
180  PRINT " "
190  NEXT I
200  PRINTER IS 1
210  END
```

```
10!  RE-STORE "DISCPRINT"
20 !ACCESS AND PRINT VANDENBERG DISC FILES
21 !
23 OPTION BASE 1
30   INTEGER T(182)
60   ASSIGN @Fill TO "VANDAT:HP82901,700,0";FORMAT OFF
70   PRINTER IS 701
80   FOR R=1 TO 5                              !SHORT PRINT *****
90   ENTER @Fill;T(*)
100  IMAGE 6D,6D,6D,6D,6D,6D,6D,6D,6D,6D
110  FOR K=1 TO 18
120  J=10*(K-1)
130  PRINT USING 100;T(1+J),T(2+J),T(3+J),T(4+J),T(5+J),T(6+J),T(7+J)
                    ,T(8+J),T(9+J),T(10+J)
140  NEXT K
150  PRINT USING "7D,7D";T(181),T(182)
160  PRINT " "
170  NEXT R
180  PRINTER IS 1
190  END
```

```
10!    RE-STORE "QA DISC"
20     !PRINT # FILES AND TIMES FOR QA DISCS
30     !
40     OPTION BASE 1
50     INTEGER Tape(183),R0,R1,I
60     DEG
70     ASSIGN @Fil0 TO "COUNTER:HP82901,700,0";FORMAT OFF
80     ASSIGN @Fil1 TO "COUNTER:HP82901,700,1";FORMAT OFF
90     ASSIGN @Fil3 TO "VANDAT:HP82901,700,0";FORMAT OFF
100    ASSIGN @Fil4 TO "VANDAT:HP82901,700,1";FORMAT OFF
110    !
120 Start:      !**** BEGIN ****
130    PRINTER IS 701
140    ENTER @Fil0,1;R0
150    ENTER @Fil1,1;R1
160    PRINT "MAX FILE # ON DISC 0 IS";R0,"MAX FILE # ON DISC 1 IS";R1
170    PRINT " "
171    Pp=0
180    !
190    PRINT "DISC 0"
200    IF R0=0 THEN 280
210      Dis=0
220      FOR I=1 TO R0
230        ENTER @Fil3,I;Tape(*)
240        GOSUB Prints
250      NEXT I
260    PRINT " "
261    Pp=0
270    !
280    PRINT "DISC 1"
290    IF R1=0 THEN 360
300      Dis=1
310      FOR I=1 TO R1
320        ENTER @Fil4,I;Tape(*)
330        GOSUB Prints
340      NEXT I
350    !
360    PRINT " "
370    PRINT "ALL DONE"
380    PRINT USING "3/"
390    PRINTER IS 1
400    GOTO 550
410    !
420 Prints:                                          !PRINT RESULTS
430         Time=Tape(182)
440         Mo=INT(Tape(181)/1000)
450         Dy=INT(Tape(181)/10)-Mo*100
460         Yr=80+Tape(181)-1000*Mo-10*Dy
470    PRINT USING "4D,"" PST"",X,DD,""/"",DD,""/"",DD,X,#";Time,Mo,Dy,Yr
480    Ztime=Time+800
490    IF Ztime<2400 THEN 520
500    Ztime=Ztime-2400
510    Dy=Dy+1
```

```
520    PRINT USING "4Z,""Z"",X,DD,X,DD,2X,#";Ztime,Dy,Mo
521    IF Pp=1 THEN 524
522    GOTO 530
524    PRINT USING """D"",D,X,""F"",3D,4X";Dis,I
525    GOTO 533
530    PRINT USING """D"",D,X,""F"",3D,4X,#";Dis,I
533    Pp=Pp+1
534    IF Pp=2 THEN Pp=0
540    RETURN
550                     END
```

```
10!    RE-STORE "VANQA"
20     !ON SITE QA FOR VANDENBERG DATA
30     !
40     OPTION BASE 1
50     DIM Wind(30),Dir(30),Temp(10),Dtem(16),Dew(5),Sigv(30),Sigth(30)
60     INTEGER Tape(183),Wctr(30)
70     INTEGER I,J,K
80     DEG
90     ASSIGN @Fil0 TO "VANDAT:HP82901,700,0";FORMAT OFF
100    ASSIGN @Fil1 TO "VANDAT:HP82901,700,1";FORMAT OFF
110    !
120    ! ***** SETTING UP THE SENSOR CALIBRATIONS *****
130    Wss=.04248
150    !Wind Direction is already in Degrees from the DAS
160    Tes=.05859
170    Dts=.02441
180    Dtm=-10
190    Tds=.097656
200    Tdm=-80
210    Vis=.02441
230    Bps=.07988
240    Bpm=909.2
250    Sws=.00097656
270    !          ***  FORMATS FOR PRINT  ***
280    !
290 Aa:  IMAGE 3Z,2X,3D,2X,3D,3X,3Z,3X,3D.D,4X,3D.D
300 Bb:  IMAGE    5X,3D,2X,3D,3X,3Z,3X,3D.D,4X,3D.D
310 Cc:  IMAGE    5X,3D,2X,3D,3X,3Z,3X,3D.D
320 Dd:  IMAGE 3Z,2X,3D,2X,3D,3X,3Z,3X,3D.D
330 Ee:  IMAGE 3Z,2X,3D,2X,3D,3X,3Z,3X,3D.D,4X,3D.D,4X,3D.D,7X,2D.D,3X,4D.D
340 Ff:  IMAGE    5X,3D,2X,3D,3X,3Z,3X,3D.D,4X,3D.D,4X,3D.D
350 Gg:  IMAGE 3Z,2X,3D,                           23X,3D.D
360      !
370    F=1
380    I=1
390    D=0
400    !
410 Start:      !**** BEGIN PRINT LOOP ****
420    PRINTER IS 1
430    PRINT "MAXIMUM FILE NUMBER IS 700"
440    INPUT "INPUT START AND END FILES (X,X) (CONT)",Ra,Rb
450    INPUT "INPUT DISC # (0 OR 1)  (CONT)",Dis
460    IF Dis<>0 AND Dis<>1 THEN 450
470    PRINTER IS 701
480    FOR R=Ra TO Rb STEP 24
481      MAT Wind= (0)
482      MAT Dir= (0)
483      MAT Wctr= (0)
484      MAT Temp= (0)
485      MAT Dtem= (0)
486      MAT Dew= (0)
487      MAT Sigv= (0)
488      MAT Sigth= (0)
```

```
490         ON END @Fil0 GOTO Switch
500         ON END @Fil1 GOTO Switch
510    !
520         DISP "START ";Ra,"WORKING ON ";R,"END ";Rb
530         IF Dis=1 THEN 560
540         ENTER @Fil0,R;Tape(*)
550         GOTO 580
560         ENTER @Fil1,R;Tape(*)
570         !
580         Time=Tape(182)
590         Mo=INT(Tape(181)/1000)
600         Dy=INT(Tape(181)/10)-Mo*100
610         Yr=80+Tape(181)-1000*Mo-10*Dy
620    IF Tape(183)=15 THEN 660
630        PRINT "BAD FILE",Mo;"/";Dy;"/";Yr,Time;"PST"
640        PRINT USING "/"
650        GOTO Start
660    FOR J=1 TO 30
670        Wind(J)=Wss*Tape(J)
671        Dir(J)=Tape(J+30)/10
672        IF Wind(J)<.1 THEN 676
674        Sigv(J)=Wss*Tape(J+60)
675        Sigth(J)=ATN(Sigv(J)/Wind(J))
676        Wctr(J)=Tape(J+90)
680    NEXT J
770    FOR J=121 TO 130
780        Temp(J-120)=120-Tes*Tape(J)
790    NEXT J
800    FOR J=131 TO 146
810        Dtem(J-130)=Dtm+Dts*Tape(J)
820    NEXT J
830    FOR J=147 TO 151
840        Dew(J-146)=Tdm+Tds*Tape(J)
850    NEXT J
860    Visi=49.61/(Tape(152)+1)
870    Bar=Bpm+Bps*Tape(153)
880    Sw=Swm+Sws*Tape(154)
890    !
891    PRINT USING "4D,"" PST"",4X,DD,2X,DD,2X,DD";Time,Dy,Mo,Yr
892    Ztime=Time+800
893    IF Ztime<2400 THEN 900
894    Ztime=Ztime-2400
895    Dy=Dy+1
900 Aaa:  IMAGE "4Z,"" ZULU"",3X,DD,2X,DD,2X,DD,5X,""15min ave"",#"
901    PRINT USING Aaa;Ztime,Dy,Mo,Yr
910    PRINT USING "3X,""DISC-"",D,3X,""FILE-"",3D";Dis,R
920    PRINT USING "/"
930    PRINT USING """NORTH VANDENBERG"",/"
940    PRINT "SITE  LVL  WS    WD    SIGMA   T-6'T"
950    !
960    PRINT USING Aa;52,12,Wind(3),Dir(3),Sigth(3),Temp(1)   !tower 052
970    PRINT USING Bb;54,Wind(11),Dir(11),Sigth(11),Dtem(1)
980    PRINT USING Aa;102,12,Wind(6),Dir(6),Sigth(6),Temp(6)  !tower 102
990    PRINT USING Bb;54,Wind(16),Dir(16),Sigth(16),Dtem(5)
```

```
1000  PRINT USING Cc;102,Wind(21),Dir(21),Sigth(21)
1010  PRINT USING "/"
1020  !
1030  PRINT USING """SOUTH VANDENBERG"",/"
1040  PRINT "SITE  LVL  WS    WD    SIGMA    T-6T    DPT     VIS06    BAR06"
1050  !
1060  PRINT USING Dd;9,12,Wind(1),Dir(1),Sigth(1)                    !twr 009
1070  PRINT USING Dd;14,12,Wind(2),Dir(2),Sigth(2)                   !twr 014
1080  PRINT USING Aa;54,12,Wind(4),Dir(4),Sigth(4),Temp(2)           !twr 054
1090  PRINT USING Bb;54,Wind(12),Dir(12),Sigth(12),Dtem(2)
1100  PRINT USING Gg;55,12,Temp(3)                                   !twr 055
1110  PRINT USING Bb;40,Wind(13),Dir(13),Sigth(13),Dtem(3)
1120  PRINT USING Aa;56,40,Wind(14),Dir(14),Sigth(14),Temp(4)        !twr 056
1130  PRINT USING Aa;101,12,Wind(5),Dir(5),Sigth(5),Temp(5)          !twr 101
1140  PRINT USING Bb;54,Wind(15),Dir(15),Sigth(15),Dtem(4)
1150  PRINT USING Aa;103,12,Wind(7),Dir(7),Sigth(7),Temp(7)          !twr 103
1160  PRINT USING Bb;54,Wind(17),Dir(17),Sigth(17),Dtem(6)
1170  PRINT USING Aa;200,12,Wind(8),Dir(8),Sigth(8),Temp(8)          !twr 200
1180  PRINT USING Bb;54,Wind(18),Dir(18),Sigth(18),Dtem(7)
1190  PRINT USING Cc;102,Wind(22),Dir(22),Sigth(22)
1200  PRINT USING Bb;204,Wind(26),Dir(26),Sigth(26),Dtem(12)
1210  PRINT USING Dd;299,108,Wind(24),Dir(24),Sigth(24)              !twr 299
1220  PRINT USING Aa;300,12,Wind(9),Dir(9),Sigth(9),Temp(9)          !twr 300
1230  PRINT USING Bb;54,Wind(19),Dir(19),Sigth(19),Dtem(8)
1240  PRINT USING Bb;102,Wind(23),Dir(23),Sigth(23),Dtem(10)
1250  PRINT USING Bb;204,Wind(27),Dir(27),Sigth(27),Dtem(13)
1260  PRINT USING Bb;300,Wind(29),Dir(29),Sigth(29),Dtem(15)
1270                                                                 !twr 301
1280 PRINT USING Ee;301,12,Wind(10),Dir(10),Sigth(10),Temp(10),Dew(1),Visi,Bar
1290  PRINT USING Ff;54,Wind(20),Dir(20),Sigth(20),Dtem(9),Dew(2)
1300  PRINT USING Ff;102,Wind(25),Dir(25),Sigth(25),Dtem(11),Dew(3)
1310  PRINT USING Ff;204,Wind(28),Dir(28),Sigth(28),Dtem(14),Dew(4)
1320  PRINT USING Ff;300,Wind(30),Dir(30),Sigth(30),Dtem(16),Dew(5)
1330  PRINT USING "49X,5A,DDD.DD";"SW = ",Sw
1340  PRINT USING "/////"
1350  !
1360  NEXT R
1370  GOTO Start
1380 Switch:                         !CHANGE DISCS
1390  PRINT "FILE NUMBER TOO LARGE"
1400  GOTO Start
1510  END
```

```
10! RE-STORE "QA"
20  !QA FOR VANDENBERG DISC FILES OR FORTRAN        3/19/84
30  !
40  OPTION BASE 1
50  DIM Sens$(94)[9],S$[9],C$[2],T$[3],Tow$(13)[3],Twr$(13)[3]
60  DIM Ddd(9,183),Dafi(7,183),Dah(5,183),Da(9,183),Daf(9,183)
61  DIM Sensr$(94)[9],Dat(183)
70  INTEGER Sens(94),A,B,C,D,I,J,K,L,Tape(183),T(183),Sensor(13,24)
80  INTEGER Cws(30),Raws(30,10,10),Rawd(30,10,10),Sigx(30,10,10)
90  INTEGER Cwd(30),Siws(30,10,10),Siwd(30,10,10),Rafws(30),Rafwd(30)
100 INTEGER Conws(3,30),Conwd(3,30),Cons(3,31),Sigwd(3,30,10,10)
110 INTEGER Rws(3,30,10,10),Rwd(3,30,10,10),Rs(3,31,10,10),Rfwd(3,30)
111 INTEGER Sigy(30,10,10),Rfws(3,30),Rfs(3,31)
120 INTEGER Wsmm,Wsmi,Wdmm,Wdmi,Temm,Temi,Dtmm,Dtmi,Tdmm,Tdmi,R(10,16)
130 INTEGER Rtemp,Rtem99,Sen,Set,Min,Mit,Hon,Hot,Minn,Mint,Large,Maxim
140 INTEGER Seclrg,Secmax,Minlrg,Minmax,Fiflrg,Fifmax,Horlrg,Hormax
150 INTEGER Ze,On,Tw,Th,Fo,Fi,Si,Et,Ni,Tn,El,Twt,Thi,Six,Nin,Nn,Hun
160 INTEGER Ofi,Ofo,Oet,Nnn,Offo,Fif,Fn,Ss,Ff,Oth,Otho,Ofs,Ofsv,Now
170 INTEGER Aa,Bb,Jn,Jo,Js,Jt,Rw,Tt,Non,Offi,Jtw,Ennd,Which,Badfile
180 INTEGER Ns,Nf,Nfi,Nh,Nso,Nsm,Nsn,Nst,Num,Numo,Numm,Numn,Numt,Sio
190 INTEGER Ftn,Stn,Twl,Oety,Tsty,Rn,Rn9,Bad,Was,Tho,Otw,Oto,Svo(30)
210 !
220 GOSUB Setup
230 !
240 !######################## INPUT DATA SORTING ####################
250 PRINTER IS 701
300 A=Tn
310 B=Tn
320 C=Et
330 D=Si
340 Ns=Tn
350 Nf=Tn
360 Nfi=Et
370 Nh=Si
380 X10:    GOSUB Loadd
390    IF A=Tn THEN X30
400    IF A=On THEN Fifsec                    ! 15sec
410    IF Ennd=On THEN X81
420      GOTO X30
430 X20: FOR I=On TO Et
440       FOR J=On TO Oet
450          Da(Tn-I,J)=Da(Ni-I,J)
460       NEXT J
470      NEXT I
480      A=Tw
490 X30: IF B=Tn THEN X50                     ! 5min
500      IF B=On THEN Fivmin
510      IF Ennd=On THEN Fivmin
520      GOTO X50
530 X40: FOR I=On TO Et
540       FOR J=On TO Oet
550          Daf(Tn-I,J)=Daf(Ni-I,J)
560       NEXT J
570      NEXT I
```

```
580        B=Tw
590 X50:  IF C=Et THEN X70                        ! 15min
600        IF C=On THEN Fifmin
610        IF Ennd=On THEN Fifmin
620        GOTO X70
630 X60:  FOR I=On TO Si
640         FOR J=On TO Oet
650           Dafi(Et-I,J)=Dafi(Sv-I,J)
660         NEXT J
670        NEXT I
680        C=Tw
690 X70:  IF D=Si AND Ennd=On THEN X8000   ! 1hour
700        IF D=Si THEN X10
710        IF D=On THEN Hour
720        IF Ennd=On THEN Hour
730        GOTO X10
740 X80:  FOR I=On TO Fo
750         FOR J=On TO Oet
760           Dah(Si-I,J)=Dah(Fi-I,J)
770         NEXT J
780        NEXT I
790        D=Tw
800      IF Ennd=On THEN X8000
810      GOTO X10
820    !
830    !############# QUALITY ANALYSIS SUBROUTINES #####################
840    !     ################# 15 sec DATA ###################
850 Fifsec:   !
860           Ns=Ns-On
880           IF Ennd=On THEN X81
890           IF Ns<Fi THEN Ns=Fi
900 X81:      Nso=Ns+On
910           Nst=Ns+Tw
920           Nsm=Ns+Fo
930           IF Nsm>Ni THEN Nsm=Ni
940           Nsn=Ns-Fo
950           IF Nsn<On THEN Nsn=On
960           IF Ns=Ni THEN X82
970      Tmt=On                                         ! DELTA TIME TEST
980      Hon=Da(Ns,182)-INT(Da(Ns,182)/Hun)*Hun
990      Hot=Da(Nso,182)-INT(Da(Nso,182)/Hun)*Hun
1000     Min=INT(Da(Ns,181)/Hun)
1010     Mit=INT(Da(Nso,181)/Hun)
1020     Sen=Da(Ns,181)-Min*Hun
1030     Set=Da(Nso,181)-Mit*Hun
1040     Secn=Sen+Six*Min+3600.*Hon
1050     Sect=Set+Six*Mit+3600.*Hot
1060     IF Hon=Ze AND Hot=Twt THEN Secn=Secn+86400
1061     Sss=Secn-Sect
1070     IF Sss>Seclrg THEN Tmt=1.5                          ! TOO MUCH TIME
1080     IF Sss>Secmax THEN Tmt=Fi
1081     IF Sss<15 THEN Tmt=Fi
1090     !
1100 X82: FOR J=On TO Thi                                    ! DATA LOOP
```

```
1110        Jt=J+Thi
1120        Js=J+Six
1130        Jn=J+Nin
1140        Jo=J+Otw
1150        Jf=J+Ofi
1160        IF Da(Ns,J)=Nnn THEN X165
1170        Aa=Ze
1180      Xxx=Da(Ns,J)
1190      Yyy=Da(Ns,Jt)
1200      Wsnow=SQR(Xxx*Xxx+Yyy*Yyy)
1210      GOSUB Windir
1220      Wdnow=Wdd
1230      IF Ns=Ni THEN X85
1240      IF Da(Nso,J)=Nnn THEN X83
1250      Xxx=Da(Nso,J)
1260      Yyy=Da(Nso,Jt)
1270      Wswas=SQR(Xxx*Xxx+Yyy*Yyy)
1271           Rw=INT(Wswas/Th)+On
1272           IF Rw>Tn THEN Rw=Tn
1280      GOSUB Windir
1290      Wdwas=Wdd
1300      Wddiff=ABS(Wdnow-Wdwas)
1310      IF Wddiff>Oety THEN Wddiff=Tsty-Wddiff
1320      IF Ns=Et THEN X83
1330      IF Da(Nst,J)=Nnn THEN X83
1340      Xxx=Da(Nst,J)
1350      Yyy=Da(Nst,Jt)
1360      Wsthn=SQR(Xxx*Xxx+Yyy*Yyy)
1370      GOSUB Windir
1380      Wdthn=Wdd
1390      Wddif2=ABS(Wdnow-Wdthn)
1400      IF Wddif2>Oety THEN Wddif2=Tsty-Wddif2
1410   !
1420   !                ****** CONSTANT TEST ******
1430 X83: FOR I=Nso TO Nsm                                        ! WS
1440        When=SQR(Da(I,J)*Da(I,J)+Da(I,Jt)*Da(I,Jt))
1450        IF ABS(Wsnow-When)>.04 THEN X85
1460        Aa=Aa+On
1470      NEXT I
1480 X85: IF Ns=Tw THEN X87
1490      FOR I=Ns-On TO Nsn STEP -On
1491      IF Ennd=On AND I=Tw THEN X87
1500       Wlbe=SQR(Da(I,J)*Da(I,J)+Da(I,Jt)*Da(I,Jt))
1510       IF ABS(Wsnow-Wlbe)>.04 THEN X87
1520       Aa=Aa+On
1530      NEXT I
1540 X87: IF Aa<Fo THEN X88
1550      Cws(J)=Cws(J)+On
1561      GOTO X165
1570 X88:   Aa=Ze                                                 ! WD
1580        IF Ns=Ni THEN X90
1590        FOR I=Nso TO Nsm
1600          Xxx=Da(I,J)
1610          Yyy=Da(I,Jt)
```

```
1620            GOSUB Windir
1630            Rrr=ABS(Wdnow-Wdd)
1640            IF Rrr>180 THEN Rrr=360-Rrr
1650            IF Rrr>.01 THEN X90
1660            Aa=Aa+On
1670          NEXT I
1680 X90:     IF Ns=Tw THEN X100
1690          FOR I=Ns-On TO Nsn STEP -On
1691          IF Ennd=On AND I=Tw THEN X100
1700            Xxx=Da(I,J)
1710            Yyy=Da(I,Jt)
1720            GOSUB Windir
1730            Rrr=ABS(Wdnow-Wdd)
1740            IF Rrr>180 THEN Rrr=360-Rrr
1750            IF Rrr>.01 THEN X100
1760            Aa=Aa+On
1770          NEXT I
1780 X100:  IF Aa<Fo THEN X110
1800        Cwd(J)=Cwd(J)+On
1810        GOTO X165
1820    !
1830    ! ******* RATE TEST *******
1840 X110:                     IF Ns=Ni OR Tmt=Fi THEN X220
1850                           IF Da(Nso,J)=Nnn THEN X220
1860                           Bad=Ze
1881            Rrr=ABS(Wsnow-Wswas)
1890            IF Rrr<R(Rw,9)*Tmt THEN X120                            ! WS
1891            IF Ns=Et THEN X130
1892            IF ABS(Wsnow-Wsthn)<.6*Wsthn THEN X140                 ! TWO BACK
1893 X130:      Bad=On
1894            Rafws(J)=Rafws(J)+On
1896            GOTO X140
1897 X120:      IF Tmt<>On THEN X140
1898            Rrr=Rrr*Ni/R(Rw,1)
1900            FOR K=On TO Ni
1910              IF Rrr<K THEN X125
1920            NEXT K
1940 X125:      Raws(J,Rw,K)=Raws(J,Rw,K)+On
1950             !
2010 X140:         IF Wddiff<R(Rw,13)*Tmt THEN X150                    ! WD
2011               IF Ns=Et THEN X160
2012               IF Wddif2<1.3*R(Rw,13) THEN X164                    ! TWO BACK
2014 X160:         Rafwd(J)=Rafwd(J)+On
2017               GOTO X165
2019 X150:         IF Tmt<>On THEN X164
2020               Rrr=Wddiff*Ni/R(Rw,5)
2022               FOR K=On TO Ni
2030                 IF Rrr<K THEN X155
2040               NEXT K
2060 X155:         Rawd(J,Rw,K)=Rawd(J,Rw,K)+On
2120       !
2130       !****** 999 OUTPUT DATA ******
2140 X164:         IF Bad<>On THEN X170
2150 X165:         Tape(J)=Nnn
```

```
2160                 Tape(Jt)=Nnn
2170                 Tape(Js)=Nnn
2180                 Tape(Jn)=Nnn
2190                 Tape(Jo)=Nnn
2200                 Tape(Jf)=Da(Ns,Jf)
2210                 GOTO X230
2220       !
2230       !      ****** SIGMA TEST ******
2240 X170: IF Tmt<>On OR Da(Nso,Jf)<Tw THEN X220
2241       Da(Nso,Jo)=100
2250       Sxs=Da(Nso,Js)*Da(Nso,Js)
2260       Sys=Da(Nso,Jn)*Da(Nso,Jn)
2270       Xbs=Da(Nso,J)*Da(Nso,J)
2280       Ybs=Da(Nso,Jt)*Da(Nso,Jt)
2290       Sigu=SQR(Sxs+Sys)
2300       Xbyb=Da(Nso,J)*Da(Nso,Jt)
2310       Sign=Da(Nso,Jf)
2320       Xyb=Wss*Wss*Da(Nso,Jo)*Da(Nso,Jo)/Sign
2330       IF Da(Nso,Jo)<Ze THEN Xyb=-Xyb
2340       Sigv=Tw*Sign*Xbyb*(Xyb-Xbyb)/(Sign-On)
2350       Sigv=SQR((Xbs*Sys+Ybs*Sxs-Sigv)/(Xbs+Ybs))
2360       Sigth=ATN(Sigv/Wswas)
2370       Rrr=Tw*ABS(Wsnow-Wswas)/Sigu                         ! WS
2380       FOR K=On TO Ni
2390         IF Rrr<K THEN X180
2400       NEXT K
2410 X180: Siws(J,Rw,K)=Siws(J,Rw,K)+On
2420       Rrr=Tw*Wddiff/Sigth                                  ! WD
2430       FOR K=On TO Ni
2440         IF Rrr<K THEN X190
2450       NEXT K
2460 X190:  Siwd(J,Rw,K)=Siwd(J,Rw,K)+On
2470        Rw=INT(ABS(Da(Ns,J)/Th))+On                       ! X-COMPONENT
2480        IF Rw>Tn THEN Rw=Tn
2490        Rrr=Tw*ABS(Da(Ns,J)-Da(Nso,J))/Da(Nso,Js)
2500        FOR K=On TO Ni
2510          IF Rrr<K THEN X200
2520        NEXT K
2530 X200:  Sigx(J,Rw,K)=Sigx(J,Rw,K)+On
2540        Rw=INT(ABS(Da(Ns,Jt)/Th))+On                      ! Y-COMPONENT
2550        IF Rw>Tn THEN Rw=Tn
2560        Rrr=Tw*ABS(Da(Ns,Jt)-Da(Nso,Jt))/Da(Nso,Jn)
2570        FOR K=On TO Ni
2580        IF Rrr<K THEN X210
2590        NEXT K
2600 X210:  Sigy(J,Rw,K)=Sigy(J,Rw,K)+On
2620   !    ****** OUTPUT DATA OK ******
2630 X220:       Tape(J)=Tn*Da(Ns,J)
2640             Tape(Jt)=Tn*Da(Ns,Jt)
2650             Tape(Js)=Hun*Da(Ns,Js)
2660             Tape(Jn)=Hun*Da(Ns,Jn)
2670             Tape(Jo)=.4251*Da(Ns,Jo)
2680             Tape(Jf)=Da(Ns,Jf)
2690   !
```

```
2700 X230:                 NEXT J
2701   !
2720   !    ****** OUTPUT DATA TO DISC ******
2730            Tape(181)=Da(Ns,181)
2740            Tape(182)=Da(Ns,182)
2750            Tape(183)=Da(Ns,183)
2760 !        OUTPUT @Tapo;Tape(*)
2810                  IF Ns>Fi THEN Fifsec
2820                  IF Ennd=Ze THEN X20
2830                  IF Ns=Tw THEN X30
2840                  GOTO Fifsec
2850 !
2860 !############### LONG AVERAGE CALCULATIONS #####################
2870 Longdata:  !
2880      IF Num=Numm THEN X400
2890      Tmt=On
2900      !
2910      Hon=INT(Ddd(Num,182)/Hun)                      ! DELTA TIME CHECK
2920      Hot=INT(Ddd(Numo,182)/Hun)
2930      Mit=Ddd(Numo,182)-Hot*Hun
2940      Min=Ddd(Num,182)-Hon*Hun
2950      Minn=Min+Six*Hon
2960      Mint=Mit+Six*Hot
2970      IF Hon=Ze AND Hot=Twt THEN Minn=Minn+Offo
2971      Mmm=Minn-Mint
2980      IF Mmm>Large THEN Tmt=1.5                      ! TOO MUCH TIME
2990      IF Mmm>Maxim THEN Tmt=Fi
2991      IF Mmm<Fi THEN Tmt=Fi
3000      !
3010 !*************** WIND-SENSOR ANALYSIS ****************
3020 X400:    FOR J=On TO Thi
3030              Jt=J+Thi
3040              Js=J+Six
3050              Jn=J+Nin
3060        !
3070        !******* WS CONSTANT TEST ********
3080        IF Ddd(Num,J)=Nnn THEN X550
3090        Aa=Ze
3100        IF Num=Numm THEN X500
3101         Rw=INT(Ddd(Numo,J)/Th)+On
3102         IF Rw>Tn THEN Rw=Tn
3110        FOR I=Numo TO Numm
3120           IF ABS(Ddd(Num,J)-Ddd(I,J))>.06 THEN X500
3130           Aa=Aa+On
3140        NEXT I
3150 X500: IF Num=Tw THEN X510
3160        FOR I=Num-On TO Numn STEP -On
3161        IF Ennd=On AND I=Tw THEN X510
3170          IF ABS(Ddd(Num,J)-Ddd(I,J))>.06 THEN X510
3180          Aa=Aa+On
3190        NEXT I
3200 X510: IF Aa<Numc THEN X520
3210        Conws(Which,J)=Conws(Which,J)+On
3220 X550: Tape(J)=Nnn
```

```
3231        GOTO X600
3240    !
3250        !******** WS RATE TEST ********
3260 X520:  IF Tmt=Fi OR Num=Numm THEN X560
3270        IF Ddd(Numo,J)=Nnn THEN X560
3300        Rrr=ABS(Ddd(Num,J)-Ddd(Numo,J))
3301        IF Rrr<R(Rw,Rn9-4)*Tmt THEN X530
3302        IF Numt>Numm THEN X540                                  ! TWO BACK
3303        IF ABS(Ddd(Num,J)-Ddd(Numt,J))<.6*Ddd(Numt,J) THEN X560
3305 X540:  Rfws(Which,J)=Rfws(Which,J)+On
3307        Tape(J)=Nnn
3308        GOTO X600
3310 X530:  IF Tmt<>On THEN X560
3311        Rrr=Rrr*Ni/R(Rw,Rn-4)
3312        FOR K=On TO Ni
3320          IF Rrr<K THEN X535
3330        NEXT K
3350 X535:  Rws(Which,J,Rw,K)=Rws(Which,J,Rw,K)+On
3421 X560:  Tape(J)=Tn*Ddd(Num,J)
3430    !
3440              !******** WD CONSTANT TEST ********
3450 X600:        IF Ddd(Num,Jt)=Nnn THEN X625
3460              Aa=Ze
3470              IF Num=Numm THEN X610
3480              FOR I=Numo TO Numm
3490                IF ABS(Ddd(Num,Jt)-Ddd(I,Jt))>.18 THEN X610
3500                Aa=Aa+On
3510              NEXT I
3520 X610:        IF Num=Tw THEN X620
3530              FOR I=Num-On TO Numn STEP -On
3531              IF Ennd=On AND I=Tw THEN X620
3540                IF ABS(Ddd(Num,Jt)-Ddd(I,Jt))>.18 THEN X620
3550                Aa=Aa+On
3560              NEXT I
3570 X620:        IF Aa<Numc THEN X630
3580              Conwd(Which,J)=Conwd(Which,J)+On
3590 X625:        Tape(Jt)=Nnn
3591              Tape(Js)=Nnn
3600              GOTO X800
3610            !
3620            !******** WD RATE TEST ********
3630 X630:      IF Num=Numm OR Tmt=Fi THEN X700
3640            IF Ddd(Numo,Jt)=Nnn THEN X700
3670            Rrr=ABS(Ddd(Num,Jt)-Ddd(Numo,Jt))
3680            IF Rrr>Oety THEN Rrr=Tsty-Rrr
3690            IF Rrr<R(Rw,Rn9) THEN X670
3691            IF Numt>Numm THEN X690                              ! TWO BACK
3692            Rr=ABS(Ddd(Num,Jt)-Ddd(Numt,Jt))
3693            IF Rr>Oety THEN Rr=Tsty-Rr
3694            IF Rr<1.3*R(Rw,Rn9) THEN X700
3696 X690:      Rfwd(Which,J)=Rfwd(Which,J)+On
3697            Tape(Jt)=Nnn
3698            Tape(Js)=Nnn
3700            GOTO X800
```

```
3701            Rrr=Rrr*Ni/R(Rw,Rn)
3702 X670:      IF Tmt<>On THEN X700
3703 X660:      FOR K=On TO Ni
3710              IF Rrr<K THEN X675
3720            NEXT K
3740 X675:      Rwd(Which,J,Rw,K)=Rwd(Which,J,Rw,K)+On
3751 X700:      Tape(Jt)=Tn*Ddd(Num,Jt)
3752            Tape(Js)=Tn*Ddd(Num,Js)
3840        !
3850        !********** WD SIGMA TEST **********
3860 X710: IF Num=Numm OR Tmt<>On THEN X800
3870       IF Ddd(Numo,Jn)<Tw THEN X800
3880       Sgm=ABS(ATN(Ddd(Numo,Js)/Ddd(Numo,J)))
3890       Sgm=ABS(Ddd(Num,Jt)-Ddd(Numo,Jt))/Sgm
3900       FOR K=On TO Ni
3910         IF Sgm<K/Tw THEN X750
3920       NEXT K
3930 X750: Sigwd(Which,J,Rw,K)=Sigwd(Which,J,Rw,K)+On
3950 X800:       Tape(Jn)=Ddd(Num,Jn)
3960                     NEXT J
3970     !
3980     ! *************** NON-WIND-SENSOR ANALYSIS ***************
3990      FOR J=Oto TO Ofo
4000      IF Ddd(Num,J)=Nnn THEN X1015
4010      !
4020           !******** CONSTANT TEST ********
4030           Aa=Ze
4040           IF Num=Numm THEN X1000
4050           FOR I=Numo TO Numm
4060              IF ABS(Ddd(Num,J)-Ddd(I,J))>.01 THEN X1000
4070              Aa=Aa+On
4080           NEXT I
4090 X1000:    IF Num=Tw THEN X1010
4100           FOR I=Num-On TO Numn STEP -On
4101              IF Ennd=On AND I=Tw THEN X1010
4110              IF ABS(Ddd(Num,J)-Ddd(I,J))>.01 THEN X1010
4120              Aa=Aa+On
4130           NEXT I
4140 X1010:    IF Aa<Numc THEN X1020
4150           Cons(Which,J-Otw)=Cons(Which,J-Otw)+On
4160 X1015:    Tape(J)=Nnn
4170           GOTO X1500
4180           !
4190              !******** RATE TEST ********
4200 X1020:       IF Num=Numm OR Tmt=Fi THEN X1060
4210              IF Ddd(Numo,J)=Nnn THEN X1060
4220              Rrr=ABS(Ddd(Num,J)-Ddd(Numo,J))
4221              IF Rrr<Rtem99 THEN X1030
4222              IF Numt>Numm THEN X1050                    ! TWO BACK
4223              IF ABS(Ddd(Num,J)-Ddd(Numt,J))<Rtem99*1.5 THEN X1060
4226 X1050:       Rfs(Which,J-Otw)=Rfs(Which,J-Otw)+On
4227              Tape(J)=Nnn
4228              GOTO X1500
4229 X1030:       IF Tmt<>On THEN X1060
```

```
4230                Rrr=Rrr*Ni/Rtemp
4231                FOR K=On TO Ni
4240                 IF Rrr<K THEN X1035
4250                NEXT K
4270 X1035:         Rs(Which,J-Otw,Rw,K)=Rs(Which,J-Otw,Rw,K)+On
4340 X1060:         Tape(J)=Hun*Ddd(Num,J)
4351 X1500:                      NEXT J
4360                !
4370     !*************** DATA OUTPUT ****************
4380             Tape(152)=Tn*Ddd(Num,152)
4390             Tape(153)=Tn*Ddd(Num,153)
4400             Tape(154)=Tn*Ddd(Num,154)
4410             FOR J=Offi TO Oet
4420                Tape(J)=Ddd(Num,J)
4430             NEXT J
4440                  RETURN
4450   !
4560   !******** WD CALCULATION SUBROUTINE ********
4570 Windir:   !
4580            Sss=ASN(Xxx/SQR(Xxx*Xxx+Yyy*Yyy))
4590            IF Yyy>=Ze THEN Wdd=(360+Sss) MOD Tsty
4600            IF Yyy<Ze THEN Wdd=Oety-Sss
4610               RETURN
4620   !################## LONG AVERAGE SORTING #######################
4630 Fivmin:                                         ! 5min Data
4640            Which=On
4650            FOR J=On TO Ni
4660            FOR I=On TO Oet
4670               Ddd(J,I)=Daf(J,I)
4680            NEXT I
4690            NEXT J
4691            IF Ennd=On THEN X1100
4700 X1090:     Nf=Nf-On
4710            IF Ennd=On THEN X1100
4720            IF Nf<Fi THEN Nf=Fi
4730 X1100:     Num=Nf
4740            Numo=Num+On
4750            Numt=Num+Tw
4760            Numm=Num+Fo
4770            IF Numm>Ni THEN Numm=Ni
4780            Numn=Num-Fo
4790            IF Numn<On THEN Numn=On
4800            Numc=Fo
4810            Large=Minlrg
4820            Maxim=Minmax
4830            Rn=Si
4831            Rn9=Ftn
4832            Rtemp=Fo
4833            Rtem99=Et
4840            GOSUB Longdata
4850 !                OUTPUT @Tapo;Tape(*)
4900                IF Nf>Fi THEN X1090
4910                IF Ennd=Ze THEN X40
4920                IF Nf=Tw THEN X50
```

```
4930              GOTO X1090
4940 Fifmin:                                          ! 15min Data
4950      Which=Tw
4960      FOR J=On TO Sv
4970      FOR I=On TO Oet
4980         Ddd(J,I)=Dafi(J,I)
4990      NEXT I
5000      NEXT J
5001      IF Ennd=On THEN X1200
5010 X1190:  Nfi=Nfi-On
5020      IF Ennd=On THEN X1200
5030      IF Nfi<Fo THEN Nfi=Fo
5040 X1200: Num=Nfi
5050      Numo=Num+On
5060      Numt=Num+Tw
5070      Numm=Num+Th
5080      IF Numm>Sv THEN Numm=Sv
5090      Numn=Num-Th
5100      IF Numn<On THEN Numn=On
5110      Numc=Th
5120      Large=Fiflrg
5130      Maxim=Fifmax
5140      Rn=Sv
5141      Rn9=Fif
5142      Rtemp=Si
5143      Rtem99=Twl
5150      GOSUB Longdata
5160 !           OUTPUT @Tapo;Tape(*)
5210             IF Nfi>Fo THEN X1190
5220             IF Ennd=Ze THEN X60
5230             IF Nfi=Tw THEN X70
5240             GOTO X1190
5250  !
5260 Hour:                                            ! 1hr Data
5270           Which=Th
5280           FOR J=On TO Fi
5290           FOR I=On TO Oet
5300             Ddd(J,I)=Dah(J,I)
5310           NEXT I
5320           NEXT J
5321           IF Ennd=On THEN X1300
5330 X1290:    Nh=Nh-On
5340           IF Ennd=On THEN X1300
5350           IF Nh<Th THEN Nh=Th
5360 X1300:    Num=Nh
5370           Numo=Num+On
5380           Numt=Num+Tw
5390           Numm=Num+Tw
5400           IF Numm>Fi THEN Numm=Fi
5410           Numn=Num-Tw
5420           IF Numn<On THEN Numn=On
5430           Numc=Tw
5440           Large=Horlrg
5450           Maxim=Hormax
```

```
5460            Rn=Et
5461            Rn9=Stn
5462            Rtemp=Et
5463            Rtem99=Fif
5470            GOSUB Longdata
5480 !                  OUTPUT @Tapo;Tape(*)
5530                IF Nh>Th THEN X1290
5540                IF Ennd=Ze THEN X80
5550                IF Nh=Tw THEN X8000
5560                GOTO X1290
5570  !
5580  !################# SETTING UP THE SYSTEM ######################
5590 Setup: !
5600         DEG
5610  DATA "009-WD12","009-WS12","014-WD12","014-WS12","052-WD12"
5620  DATA "052-WS12","054-WD12","054-WS12","101-WD12","101-WS12"
5630  DATA "102-WD12","102-WS12","103-WD12","103-WS12","200-WD12"
5640  DATA "200-WS12","300-WD12","300-WS12","301-WD12","301-WS12"
5650  DATA "052-WD54","052-WS54","054-WD54","054-WS54","055-WD40"
5651  DATA "055-WS40","056-WD40","056-WS40","101-WD54","101-WS54"
5660  DATA "102-WD54","102-WS54","103-WD54","103-WS54","200-WD54"
5670  DATA "200-WS54","300-WD54","300-WS54","301-WD54","301-WD54"
5680  DATA "102-WD102","102-WS102","200-WD102","200-WS102","300-WD102"
5690  DATA "300-WS102","299-WD102","299-WS102","301-WD102","301-WS102"
5700  DATA "200-WD204","200-WS204","300-WD204","300-WS204","301-WD204"
5720  DATA "301-WS204","300-WD300","300-WS300","301-WD300","301-WS300"
5740  DATA "052-TE","054-TE","055-TE","056-TE","101-TE","102-TE"
5750  DATA "103-TE","200-TE","300-TE","301-TE","052-DT54","054-DT54"
5760  DATA "055-DT54","101-DT54","102-DT54","103-DT54","200,DT54"
5770  DATA "300-DT54","301-DT54","300-DT102","301-DT102","200-DT204"
5780  DATA "300-DT204","301-DT204","300-DT300","301-DT300","301-DP6"
5790  DATA "301-DP54","301-DP102","301-DP204","301-DP300","301-VI"
5800  DATA "301-BP","301-SW"
5810  READ Sens$(*)
5820  DATA "009","014","052","054","055","056","101","102","103","200"
5830  DATA "299","300","301"
5840  READ Tow$(*)
5850       !                        SENSOR CONVERSIONS
5860       Wss=.04248
5870       Tem=120
5880       Tes=-.05859
5890       Dtm=-10
5900       Dts=.02441
5910       Tdm=-80
5920       Tds=.097656
5930       Bpm=909.2
5940       Bps=.07988
5950       Sws=.00097656
5960            !                   VALID LIMITS FOR DATA
5970            Wsmm=70
5980            Wsmi=0
5990            Wdmm=360
6000            Wdmi=0
6010            Temm=100
```

```
6020            Temi=30
6030            Dtmm=7
6040            Dtmi=-7
6050            Tdmm=100
6060            Tdmi=10
6080     !        ******** RATE AND 999 LIMITS ********
6081 !      RATE WS        |       RATE WD       |       999 WS        |       999 WD           RW
6083 !  15s, 5m,15m, 1h,15s, 5m,15m, 1h,15s, 5m,15m, 1h,15s, 5m,15s, 1h
6090 DATA 5, 10, 10, 15,180,180,180,180, 30, 30, 90, 90,180,180,180,180 ! 1
6091 DATA 5, 10, 10, 15,120,150,160,160, 30, 30, 90, 90,180,180,180,180 ! 2
6092 DATA 6, 10, 10, 15, 90,120,140,140, 30, 30, 30, 30,140,150,180,180 ! 3
6093 DATA 6, 10, 10, 15, 45, 90,120,120, 30, 30, 90, 90,140,150,150,180 ! 4
6094 DATA 7, 10, 10, 15, 45, 60,100,100, 30, 30, 90, 90,100,120,150,150 ! 5
6095 DATA 7, 10, 10, 15, 45, 60, 80, 80, 30, 30, 90, 90,100,120,120,150 ! 6
6096 DATA 8, 10, 10, 15, 30, 45, 60, 60, 30, 30, 90, 90, 80,100,120,150 ! 7
6097 DATA 9, 10, 10, 15, 30, 45, 60, 60, 30, 30, 90, 90, 60, 90,100,150 ! 8
6098 DATA 9, 10, 10, 15, 30, 30, 40, 60, 30, 30, 90, 90, 60, 90,100,150 ! 9
6099 DATA 10,10, 10, 15, 30, 30, 40, 60, 30, 30, 90, 90, 60, 90,100,150 !10
6100   READ R(*)
6150          !       ******** TOWER SENSOR CODES ********
6160     ! 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
6170  DATA 2,1,2,0,0,0,0,0,0,0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
6180  DATA 2,3,4,0,0,0,0,0,0,0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
6190  DATA 7,5,6,61,71,21,22,0,0,0,0,0,0,0, 0, 0, 0, 0, 0, 0, 0, 0,0
6200  DATA 7,7,8,62,72,23,24,0,0,0,0,0,0,0, 0, 0, 0, 0, 0, 0, 0, 0,0
6210  DATA 5,25,26,63,73,0,0,0,0,0,0,0,0,0, 0, 0, 0, 0, 0, 0, 0, 0,0
6220  DATA 4,27,28,64,0,0,0,0,0,0,0,0,0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
6230  DATA 7,9,10,65,74,29,30,0,0,0,0,0,0,0,0, 0, 0, 0, 0, 0, 0, 0,0
6240  DATA 9,11,12,66,75,31,32,41,42,0,0,0,0,0,0,0,0, 0, 0, 0, 0, 0,0
6250  DATA 7,13,14,67,76,33,34,0,0,0,0,0,0,0,0, 0, 0, 0, 0, 0, 0, 0,0
6260  DATA 12,15,16,68,77,35,36,43,44,51,52,82,0,0,0,0,0,0,0,0,0, 0,0
6270  DATA 3,47,48,0,0,0,0,0,0,0,0,0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,0
6280  DATA 16,17,18,69,78,37,38,45,46,53,54,80,57,58,83,85,0,0,0,0,0,0
6290  DATA 0,0,24,19,20,70,79,39,40,49,50,55,56,87,81,92,93,59
6300  DATA 60,84,86,88,89,90,91,94
6310  READ Sensor(*)
6320   !
6330                    !****** I/O SETUP ******
6340 ! OUTPUT 718;"BS(4026)"
6350 ! ASSIGN @Tapi TO 719;FORMAT OFF
6360 ! OUTPUT 724;"BS(4026)"
6370 ! ASSIGN @Tapo TO 725;FORMAT OFF
6380   ASSIGN @Tapi TO "QAOUT:INTERNAL,4,1";FORMAT OFF
6390 ! ASSIGN @Tapo TO "QAOUT:INTERNAL,4,0";FORMAT OFF
6400   !
6410   !*** DEFINE INTEGERS ***
6420   Ennd=0
6430   Badfile=0
6440   Ze=0
6450   On=1
6460   Tw=2
6470   Th=3
6480   Fo=4
6490   Fi=5
```

```
6500   Si=6
6510   Sv=7
6520   Et=8
6530   Ni=9
6540   Tn=10
6550   El=11
6551   Twl=12
6552   Ftn=14
6560   Fif=15
6561   Stn=16
6570   Twt=23
6580   Thi=30
6590   Tho=31
6600   Ff=55
6610   Fn=59
6620   Six=60
6621   Sio=61
6630   Ss=66
6640   Nin=90
6650   Non=91
6660   Nn=99
6670   Hun=100
6680   Otw=120
6690   Oto=121
6700   Oth=130
6710   Otho=131
6720   Ofs=146
6730   Ofsv=147
6740   Ofi=150
6750   Ofo=151
6760   Offi=155
6761   Oety=180
6770   Oet=183
6771   Tsty=360
6780   Nnn=999
6790   Offo=1440
6800      !*** TIME LIMITS FOR RATES ***
6810      Seclrg=17
6820      Secmax=33
6830      Minlrg=6
6840      Minmax=12
6850      Fiflrg=17
6860      Fifmax=33
6870      Horlrg=65
6880      Hormax=130
6890  !*** DELETE TOWERS AND SENSORS ***
6900  Twr$(1)="000"
6920  FOR I=1 TO 13
6930    IF Twr$(I)="000" THEN X5120
6940    FOR J=1 TO 13
6950      IF Twr$(I)=Tow$(J) THEN X5080
6960    NEXT J
6970 X5080: FOR K=2 TO Sensor(J,1)
6980            Sens(Sensor(J,K))=Nn
```

```
6990          NEXT K
7000  NEXT I
7010 X5120:       Sensr$(1)="000-000"
7040              FOR I=1 TO 94
7050                IF Sensr$(I)="000-000" THEN X5200
7060                FOR J=1 TO 94
7070                  IF Sensr$(I)=Sens$(J) THEN X5180
7080                NEXT J
7090 X5180:         Sens(J)=Nn
7100              NEXT I
7110 X5200:                      RETURN
7120  !
7130  !###################### LOAD DATA ##############################
7140 Loadd: !
7160         GOTO X5350
7170         IF Ennd=On THEN X6500
7180   !***** TEST FOR EOF *****
7190   Tt=SPOLL(718)
7200   IF BIT(Tt,4)=Ze THEN X5350
7210   OUTPUT 718;"MF(1)"
7220   Tt=SPOLL(718)
7230   IF BIT(Tt,4)=Ze THEN X5350
7240   PRINT "END OF INPUT TAPE"
7250   OUTPUT 718;"RW"
7260   Ennd=On
7270   GOTO X6500
7280   !
7290   !******** ENTER DATA ********
7300 X5350:  ENTER @Tapi;T(*)
7320         IF T(183)<>999 THEN 7350
7330         Ennd=On
7340         GOTO X6500
7350   IF T(183)=El THEN Fsec
7360   IF T(183)=Ff OR T(183)=Fif OR T(183)=Ss THEN Fmin
7370          Badfile=Badfile+On
7380          GOTO Loadd
7390  !
7400 Fsec:  !**************** 15sec DATA ************************
7410              A=A-On
7420     FOR J=On TO Six
7430     Da(A,J)=Wss*T(J)
7440     NEXT J
7450     FOR J=Sio TO Otw
7460     Da(A,J)=Wss*T(J)/Tn
7470     NEXT J
7480     FOR J=Oto TO Oet
7490     Da(A,J)=T(J)
7500     NEXT J
7510   !
7520   !******** CHECK VALIDITY OF 15sec DATA *********
7530   FOR J=On TO Thi
7540      Jt=J+Thi
7550      Js=J+Six
7560      Jn=J+Nin
```

```
7570        Jo=J+Otw
7580        Jf=J+Ofi
7590        Jtw=Tw*J
7600        IF Sens(Jtw)=Nn OR Sens(Jtw-On)=Nn THEN X5610        ! DUMMIED
7610        IF Da(A,Jf)<Tw THEN X5610                            ! NO COUNT
7620        IF Da(A,J)>Wsmm OR Da(A,J)<-Wsmm THEN X5610          ! WS VALID
7630        IF Da(A,Jt)>Wsmm OR Da(A,Jt)<-Wsmm THEN X5610
7640        GOTO X5640
7650 X5610:    Da(A,J)=Nnn                                       ! 999 DATA
7660           Da(A,Jt)=Nnn
7670           Da(A,Js)=Nnn
7680           Da(A,Jn)=Nnn
7690           Da(A,Jo)=Nnn
7700 X5640:                     NEXT J
7710                            GOTO X6500
7720   !
7730 Fmin: !**************** LONG AVERAGE DATA *****************
7740                   FOR J=On TO Oet
7750                     Dat(J)=Ze
7760                   NEXT J
7770        ! ********** CHECK VALIDITY OF WIND DATA **********
7780        FOR J=On TO Thi
7790        Jt=J+Thi
7800        Js=J+Six
7810        Jn=J+Nin
7820        Jtw=Tw*J                                                 !WS
7830          IF Sens(Jtw)=Nn OR Sens(Jtw-On)=Nn THEN X6020      !DUMMIED
7840          Dat(J)=Wss*T(J)
7850          IF Dat(J)>Wsmm OR Dat(J)<Wsmi THEN X6020             ! VALID
7860          Dat(Jt)=T(Jt)/Tn                                     ! WD
7870          IF Dat(Jt)>Wdmm OR Dat(Jt)<Wdmi THEN X6020           ! VALID
7880          Dat(Js)=Wss*T(Js)
7890          Dat(Jn)=T(Jn)
7900          IF Dat(Jn)=Ze THEN X6020                         ! ZERO COUNT
7910          GOTO X6100
7920 X6020:   Dat(J)=Nnn
7930          Dat(Jt)=Nnn
7940          Dat(Js)=Nnn
7950 X6100:                        NEXT J
7960          Dat(181)=T(181)                                  ! TIME DATA
7970          Dat(182)=T(182)
7980          Dat(183)=T(183)
7990          !
8000     ! ********** CHECK VALIDITY OF TEMPERATURE DATA **********
8010              FOR J=Oto TO Oth
8020                Dat(J)=Tem+Tes*T(J)
8030                IF Sens(J-Six)=Nn THEN X6110              ! TE DUMMIED
8040                IF Dat(J)>Temm OR Dat(J)<Temi THEN X6110      ! VALID
8050                GOTO X6120
8060 X6110:         Dat(J)=Nnn
8070 X6120:       NEXT J
8080               FOR J=Otho TO Ofs
8090                 Dat(J)=Dtm+Dts*T(J)
8100                 IF Sens(J-Six)=Nn THEN X6130             ! DT DUMMIED
```

```
8110                    IF Dat(J)>Dtmm OR Dat(J)<Dtmi THEN X6130     ! VALID
8120                    GOTO X6140
8130 X6130:             Dat(J)=Nnn
8140 X6140:           NEXT J
8150                 FOR J=Ofsv TO Ofo
8160                    Dat(J)=Tdm+Tds*T(J)
8170                    IF Sens(J-Six)=Nn THEN X6150            ! TD DUMMIED
8180                    IF Dat(J)>Tdmm OR Dat(J)<Tdmi THEN X6150     ! VALID
8190                    GOTO X6160
8200 X6150:            Dat(J)=Nnn
8210 X6160:          NEXT J
8220                    Dat(152)=49.61/(T(152)+On)           ! OTHER SENSORS
8230                    Dat(153)=Bpm+Bps*T(153)
8240                    Dat(154)=Sws*T(154)
8250    !
8260    ! ********** SORT LONG AVERAGE INPUT DATA **********
8270           IF T(183)=Fif THEN X6240
8280           IF T(183)=Ss THEN X6290
8290                     B=B-On
8300                     FOR J=On TO Oet
8310                       Daf(B,J)=Dat(J)
8320                     NEXT J
8330                     GOTO X6500
8340 X6240:          C=C-On
8350                 FOR J=On TO Oet
8360                   Dafi(C,J)=Dat(J)
8370                 NEXT J
8380                 GOTO X6500
8390 X6290:       D=D-On
8400              FOR J=On TO Oet
8410                 Dah(D,J)=Dat(J)
8420              NEXT J
8430 X6500:                RETURN
8440    !
8761 X8000:                            END
```

APPENDIX B

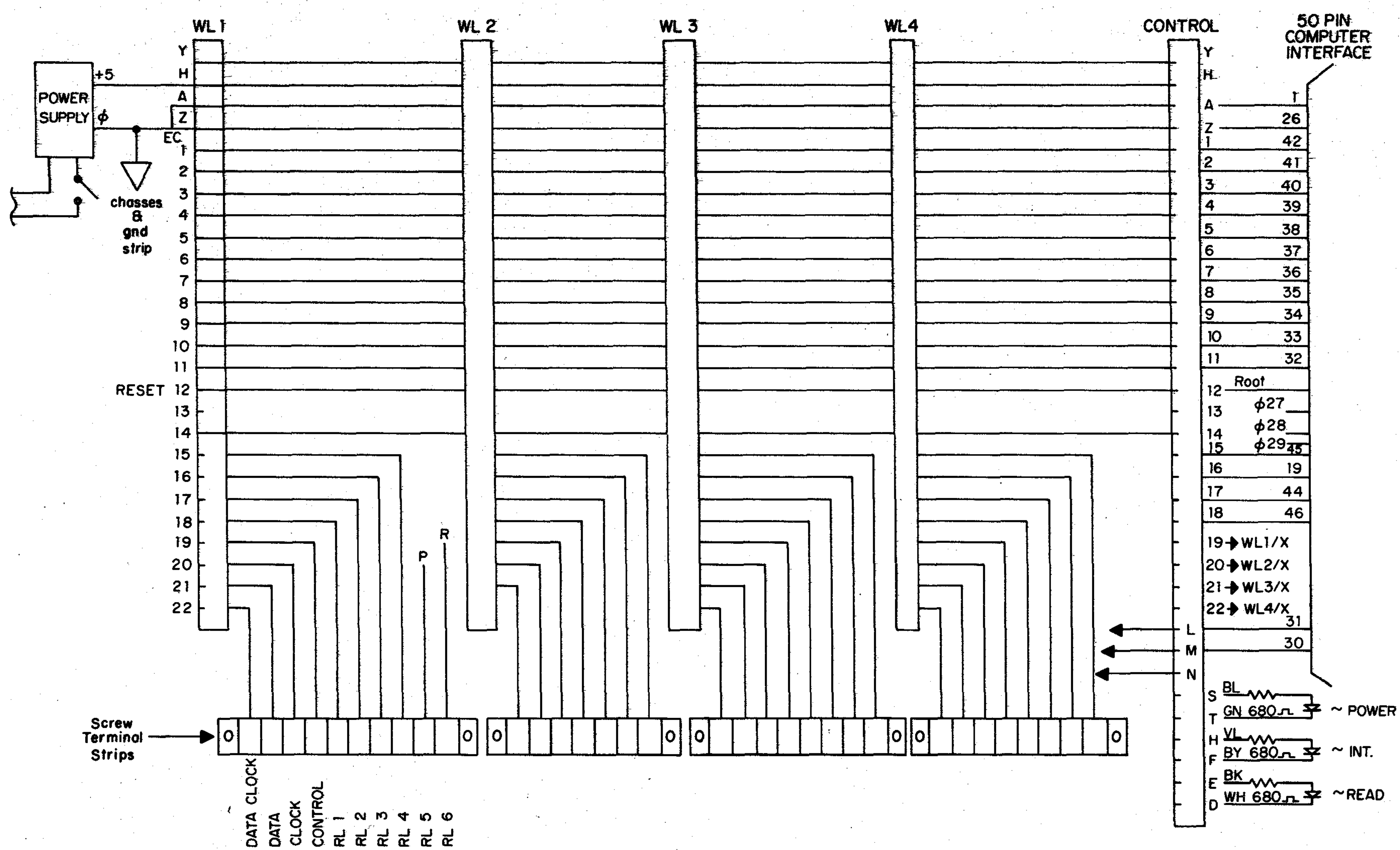
This appendix contains three circuit diagrams for the NPS data acquisition system:

1. the data capture controller,
2. the data acquisition board,
3. the system interconnection wiring.

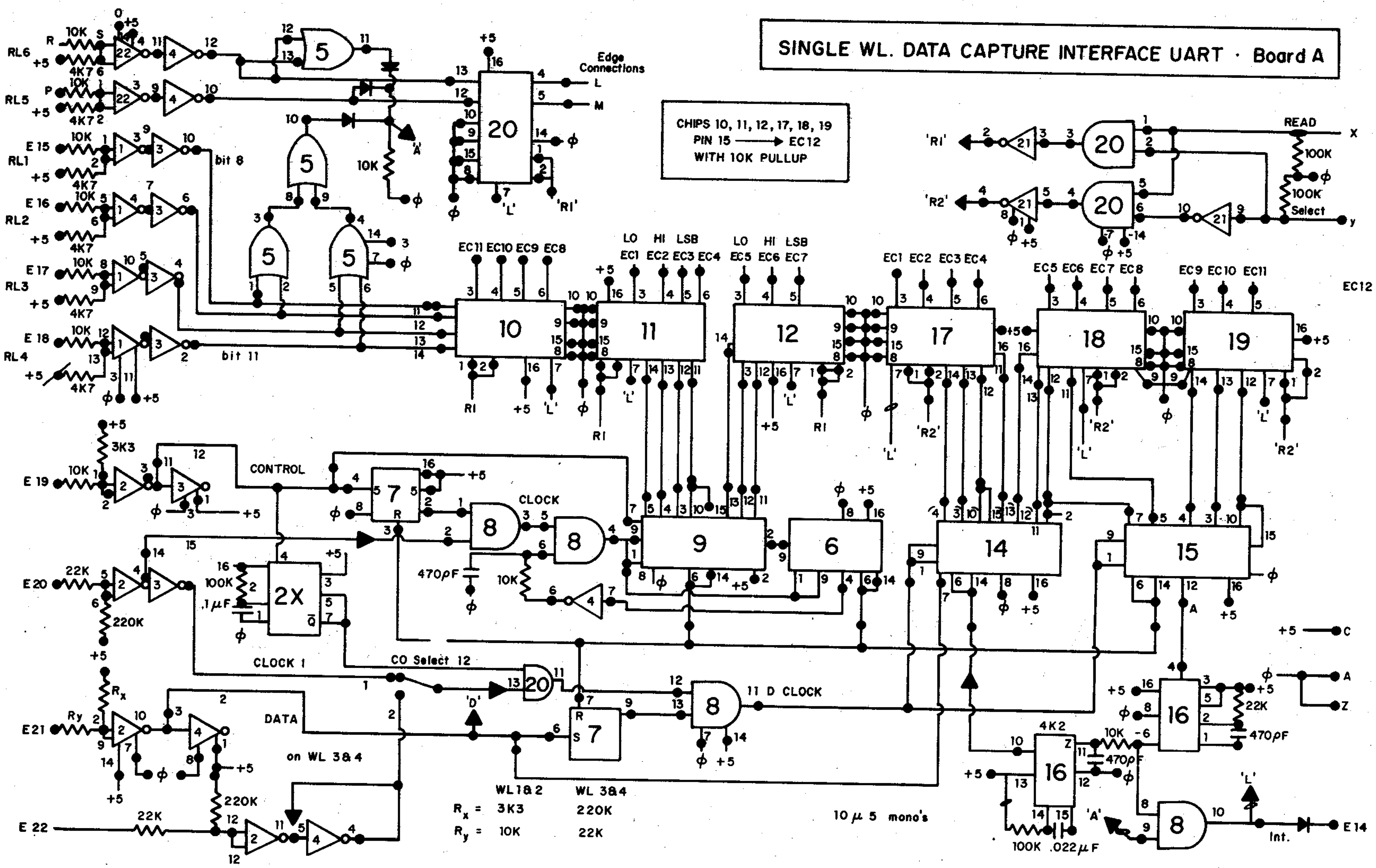
The diagrams are included without explanation, for the sake of completeness only.

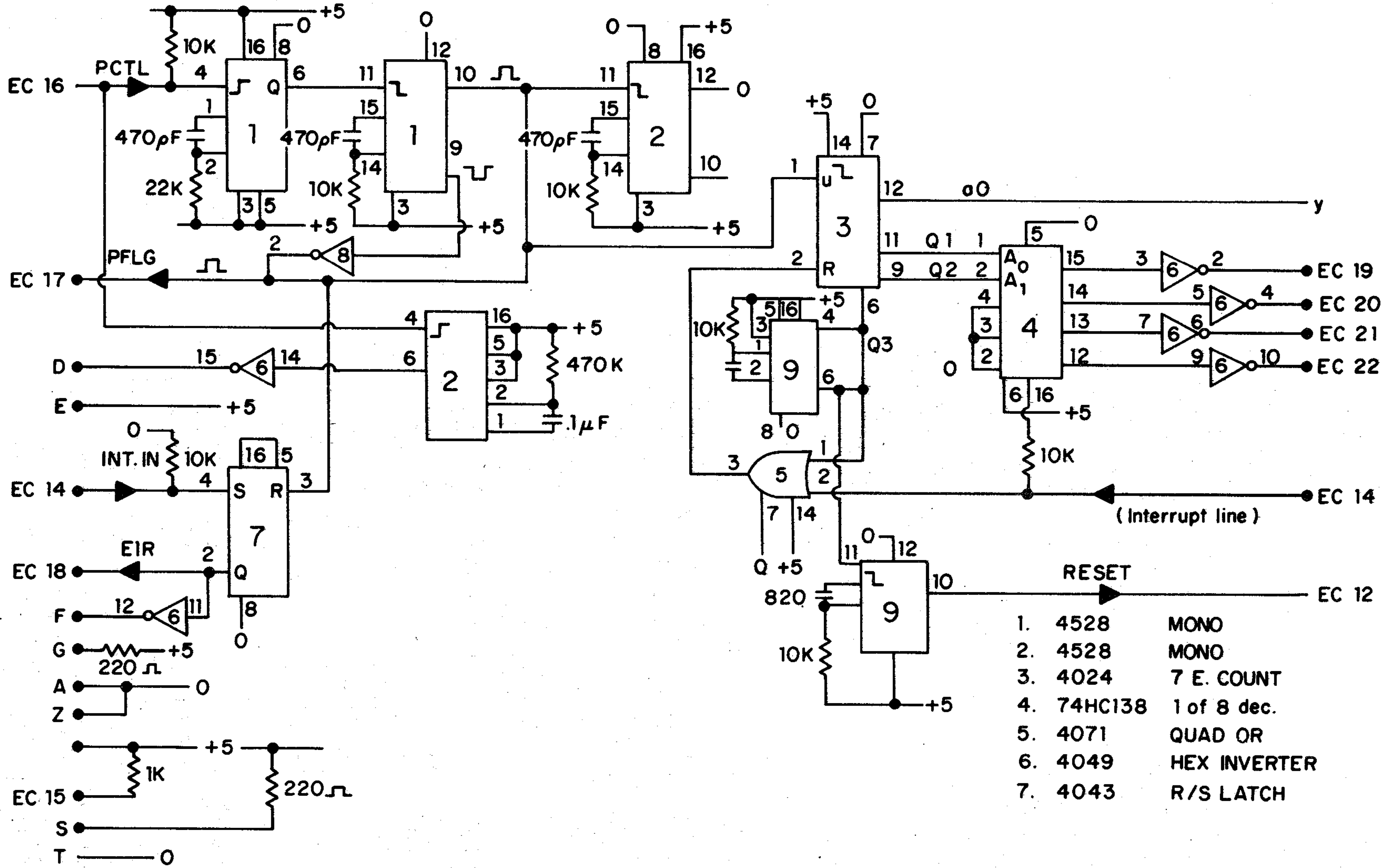
BUS A, C and Z should be insulated

EDGE CONNECTIONS



SINGLE WL. DATA CAPTURE INTERFACE UART · Board A





1.	4528	MONO
2.	4528	MONO
3.	4024	7 E. COUNT
4.	74HC138	1 of 8 dec.
5.	4071	QUAD OR
6.	4049	HEX INVERTER
7.	4043	R/S LATCH

DISTRIBUTION LIST

		No. of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93943	2
3.	Professor K.L. Davidson, Code 63Ds Department of Meteorology Naval Postgraduate School Monterey, California 93943	1
4.	Professor G. E. Schacher, Code 61Sq Department of Physics Naval Postgraduate School Monterey, California 93943	10
5.	Mr. Timothy Stanton, Code 68Sm Department of Oceanography Naval Postgraduate School Monterey, California 93943	2
6.	Lt. Col. Dennis Downing SD/CFA Headquarters Space Division P.O. Box 92960 Worldway Postal Center Los Angeles, California 90009	1
7.	LT Betty Hagan SD/CFAT Headquarters Space Division P.O. Box 92960 Worldway Postal Center Los Angeles, California 90009	1
8.	Dr. Ray Bernberg The Aerospace Corporation Space Launch Vehicle Division P. O. Box 92957 Los Angeles, California 90009	2
9.	Dr. Ron Bywater The Aerospace Corporation Space Launch Vehicle Division P.O. Box 92957 Los Angeles, California 90009	1

10. Dr. Ron Cionco 1
Atmospheric Sciences Lab
WSMR, New Mexico 80002

11. Dr. Abel Blanco 1
Atmospheric Sciences Lab
WSMR, New Mexico 80002

12. Dr. Paul Tag 25
Naval Environmental Prediction Research Facility
Monterey, California 93943

13. Mr. Sam Brand 1
Naval Environmental Prediction Research Facility
Monterey, California 93943

14. Lt. Col. William Bihner 2
WSMC/WE
Vandenberg Air Force Base, California 93437

15. Col. Syd Smith 1
WSMC/CD
Vandenberg Air Force Base, California 93437

16. Mr. Sven C. Ehrnberg 1
WSMC/RSIT
Vandenberg Air Force Base, California 93437

17. Mr. Jerry Farley 1
Federal Electric Corporation
Winds - IC 612
Vandenberg Air Force Base, California 93437

18. Dr. Steven Hannah 1
Principal Meteorologist
Environmental Research & Technology Corp
696 Virginia Road
Concord, MA 01742

19. Mr. Don Spiel 1
BDM Corporation, 1340 Munras Street
Monterey, California 93940

20. Mr. Chuck Skupniewicz 1
Department of Physics
Naval Postgraduate School
Monterey, California 93943

21. CDR S. G. Colgan | 1
Code 420 B
Office of Naval Research
800 N. Quincy Street
Arlington, VA 22217

22. MAJ Gary G. Worley | 1
Air Force Engineering and Services Center
AFESC/RDVA
Tyndall Air Force Base, Florida 32403

23. LT Bob Johnson | 2
6595th Shuttle Test Group
STG/SA
Bldg. 8500
Bandenberg, CA 93437